

Treball de Fi de Màster

Màster Universitari d'Enginyeria Industrial

Desenvolupament d'aplicacions de realitat virtual amb dispositius de posicionament i tracking d'HTC

MEMÒRIA

Autor: Riaño Pedemonte, Daniel
Director: Susin Sanchez, Toni
Convocatòria: Febrer 2019



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Aquest projecte s'ha realitzat amb la finalitat d'aprofundir en les possibilitats que ofereixen els dispositius de realitat virtual d'HTC dels que disposa el Centre de Realitat Virtual de la Facultat de Matemàtiques i Estadística.

El projecte consisteix en l'elaboració de diverses escenes 3D mitjançant el motor de jocs Unity. L'objectiu principal d'aquestes escenes és desenvolupar aplicacions de realitat virtual i realitat augmentada basades en les dades de posicionament i tracking que ofereixen els dispositius.

A la memòria s'explica la tecnologia en què es basen els dispositius i es fa un anàlisi de com obtenir les dades que genera. Posteriorment es planteja la possibilitat d'utilitzar aquestes dades per a diverses finalitats pràctiques.

Les escenes descrites reproduïxen diversos conceptes que cal entendre per a poder desenvolupar una aplicació de realitat virtual per a dispositius mòbils amb el sistema operatiu d'Android. Tots els conceptes necessaris per tal de complir l'objectiu del projecte s'expliquen al llarg d'aquest document .

Per tant, el contingut d'aquest projecte, serveix com a base per a poder desenvolupar qualsevol tipus d'aplicació de realitat virtual i augmentada per a dispositius mòbils utilitzant el posicionament i tracking dels dispositius d'HTC.

1. Sumari

RESUM	2
1. SUMARI	4
2. GLOSSARI	6
3. PREFACI	7
3.1. Origen del projecte.....	7
3.2. Requeriments previs.....	7
4. INTRODUCCIÓ	8
4.1. Objectius del projecte	8
4.2. Abast del projecte	8
5. UNITY: LA PLATAFORMA DE DESENVOLUPAMENT	9
5.1. Unity3d.....	9
5.2. C#: El llenguatge de programació.....	9
5.3. Funcionament bàsic del programa	9
6. DISPOSITIUS HTC	12
6.1. Realitat virtual	12
6.2. HMD (Head Mounted Display).....	13
6.3. Base Stations.....	14
6.4. Controladors	15
6.5. Trackers	16
6.6. Steam.....	17
6.7. Requeriments.....	18
7. IMPLEMENTACIÓ D'HTC VIVE AMB UNITY	20
7.1. Introducció: aplicacions dels dispositius i antecedents.....	20
7.2. Plugin per Unity: SteamVR	22
7.3. Desconnexió del casc.....	23
7.3.1. Motivació.....	23
7.3.2. Procediment.....	23
7.3.3. Zona de joc i assignació dels índexs.....	25
7.3.4. Creació d'una escena	26
7.4. Comunicacions entre dispositius	30
7.4.1. Limitacions dels dispositius mòbils.....	30
7.4.2. Wifi / Bluetooth.....	30

7.5. Servidor / Client	31
7.5.1. Funcionament	31
7.5.2. Implementació.....	32
7.6. Aplicacions Android.....	35
7.6.1. Com crear una app per Android.....	35
7.6.2. Requeriments Android	36
7.7. UI: Interfície de l'usuari.....	38
7.8. Realitat Virtual	41
7.8.1. Aplicació VR.....	41
7.9. Realitat Augmentada.....	44
7.9.1. Estat de l'art	44
7.9.2. Funcionament AR	46
7.9.3. Obtenció de la imatge del dispositiu.....	46
7.9.4. Implementació AR.....	47
7.9.5. Aplicació AR.....	49
7.10. Selecció i posicionament d'objectes.....	50
7.10.1. Aplicació ARposition: selecció i posicionament d'objectes.....	51
8. APLICACIONS MULTIUSUARI	54
9. CAS D'ÚS: INSTRUCCIONS DE MUNTATGE	56
9.1. Motivació	56
9.2. Objectiu	56
9.3. Aplicació ARinstructions	57
9.4. Aplicacions pràctiques.....	58
10. PLANIFICACIÓ DEL PROJECTE	59
10.1. Gestió del temps.....	59
10.2. Diagrama de Gantt.....	59
11. PRESSUPOST	63
CONCLUSIONS	65
AGRAÏMENTS	66
BIBLIOGRAFIA	67
Referències bibliogràfiques.....	67
Bibliografia complementària.....	68

2. Glossari

- Demo: Abreviació de la paraula demostració. Es refereix a una escena creada amb l'objectiu de mostrar el funcionament d'algun concepte.
- SDK: Software Development Kit (Kit de desenvolupament de software).
- Script: Fitxer de Unity que conté codi de programació en llenguatge C#.
- Plugin: Extensió externa al Software.
- RV: Realitat Virtual.
- Loop: Bucle.
- Game Loop: Bucle mitjançant el qual es formen els videojocs.
- Frame: Imatge instantània. Per a cada instant que es visualitza una imatge es realitzen tots els càlculs necessaris per actualitzar l'escena.
- PC: Personal Computer (ordinador).
- fps: Frames per segon. Mesura de la freqüència a la que es visualitzen els frames.
- GameObject: Element que forma part d'una escena de Unity.
- ETSEIB: Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.
- Tracker: dispositiu de seguiment.
- Headset: dispositiu per a visualitzar una escena virtual. Normalment en format de casc o d'ulleres.
- Lighthouse: tecnologia de posicionament.
- CRV: centre de realitat virtual.
- HMD : Head Mounted Display. Es refereix al headset d'HTC.

3. Prefaci

3.1. Origen del projecte

Aquest projecte forma part d'una generació de projectes de l'Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB) impulsada pel professor i director d'aquest projecte Toni Susín, i amb la col·laboració del Centre de Realitat Virtual de la Facultat de Matemàtiques i Estadística (FME). Aquest gran projecte es va iniciar el 2016 i pretén crear una comunitat de desenvolupament i recerca en l'àmbit de la realitat virtual. Es proposa estudiar els diferents dispositius dels que disposa aquesta tecnologia per a crear noves aplicacions. En el cas d'aquest projecte en concret, s'ha estudiat la implementació dels dispositius d'HTC, amb l'objectiu de comprendre com funcionen, com adquirir les dades que proporciona i com utilitzar-les per al desenvolupament d'aplicacions diverses.

Per altra banda, es pretén crear una sèrie de documents i arxius, tant escrits com escenes demostratives (on s'inclou el present document) per tal de facilitar el desenvolupament d'un possible futur projecte basat en aquests dispositius o la combinació d'aquests amb d'altres, implementats per diferents estudiants de la pròpia universitat.

Personalment, he desenvolupat un projecte amb el Centre de Realitat Virtual amb anterioritat, estudiant les possibilitats que ofereix el dispositiu Kinect V2. La programació i el món 3D és un àmbit en el qual m'agrada treballar i em motiva a trobar solucions mitjançant noves eines, per resoldre els problemes que van sorgint. A més, l'aprenentatge de diferents tipus d'eines per aconseguir l'objectiu del projecte – com, per exemple, la programació d'un llenguatge informàtic o el domini d'un software com Unity – genera una enorme quantitat de possibilitats molt interessants de cara al futur i fan el projecte més atractiu.

3.2. Requeriments previs

Per a la realització d'aquest projecte són necessaris conceptes adquirits de diverses assignatures, tant del Grau en Enginyeria en Tecnologies Industrials com del Màster Universitari en Enginyeria Industrial. Els coneixements més importants són els que es refereixen a la programació, ja que durant aquest projecte la principal tasca ha estat la de programar en llenguatge C#. Per tant, és necessari haver assolit els coneixements bàsics de qualsevol llenguatge de programació. Addicionalment a la programació, també és necessari un domini del motor de jocs Unity i un coneixement de la seva interfície gràfica. El fet d'haver realitzat anteriorment un projecte en un camp de recerca relacionat, ha facilitat molt la feina a l'hora d'assolir els objectius, permeten centrar-se en altres aspectes, a part de la programació i el domini del software.

4. Introducció

4.1. Objectius del projecte

L'objectiu principal del projecte és desenvolupar aplicacions de realitat virtual i augmentada utilitzant els dispositius d'HTC. Aquests dispositius faciliten la informació de posicionament i tracking imprescindible per situar a l'usuari en una escena virtual i permeten la interacció amb aquesta.

Per a crear les aplicacions, primerament cal entendre com funciona la tecnologia d'HTC, com obtenir les dades de posicionament i com utilitzar-les per a crear escenes virtuals.

També s'ha marcat com a objectiu definir un mètode de comunicació entre dispositius, per a poder reproduir les aplicacions des de dispositius mòbils amb una connexió sense fils.

Per a cada concepte important, s'ha creat una aplicació demostrativa que explica el funcionament de cada tipus d'implementació.

4.2. Abast del projecte

El projecte consta de diverses activitats que s'han dut a terme per assolir els objectius del projecte.

Inicialment hi ha hagut un període d'introducció, en el que s'han analitzat els dispositius d'HTC, s'ha estudiat com funciona la seva tecnologia i com es gestionen les dades de posicionament que generen utilitzant el software de Unity.

Les aplicacions desenvolupades s'han centrat en els trackers, els dispositius més nous d'HTC que permeten posicionar i realitzar un seguiment de l'usuari.

Per a visualitzar l'escena s'ha prescindit del casc de realitat virtual. S'han utilitzat dispositius mòbils per a aquest fi. S'ha definit un sistema de comunicacions del tipus Servidor/Client per a la transmissió de dades entre l'ordinador i els dispositius mòbils.

Una vegada definit el sistema de comunicacions, s'han creat aplicacions demostratives de realitat virtual i augmentada per a la plataforma Android. Aquestes escenes permeten entendre com s'han de crear aquest tipus d'aplicacions i les possibilitats que ofereixen.

Finalment s'ha estudiat la opció de gestionar més d'un dispositiu simultàniament i s'ha creat una aplicació per a un cas pràctic a partir de la metodologia descrita durant tot el projecte.

5. Unity: La plataforma de desenvolupament

5.1. Unity3d

Unity Technologies és una empresa danesa fundada el 2004, que pretén proporcionar un motor de videojocs econòmic i accessible al major nombre de persones. Es va idear per a desenvolupadors independents que no disposen de mitjans per crear un motor propi. És per això, que la seva interfície gràfica és senzilla, molt intuïtiva i, a més, la seva estructura permet minimitzar la quantitat de codi necessari per realitzar diverses accions. Això simplifica molt el desenvolupament d'aplicacions per a gent no experta en aspectes més tècnics, com la programació, i el fet de ser multi plataforma augmenta les possibilitats de desenvolupament. Una aplicació, un cop creada, pot ser compilada en diferents plataformes suportades per Unity: des de la plataforma per Xbox fins a mòbils amb sistema operatiu iOS o Android.

5.2. C#: El llenguatge de programació

El llenguatge de programació de Unity és el C#, un llenguatge orientat a objectes que deriva del C i el C++ i forma part de la plataforma de desenvolupament d'aplicacions de Microsoft .NET.

L'entorn de desenvolupament integrat (IDE) del projecte, ha estat Visual Studio 2015, que consta de diversos recursos tals com un editor de codi font o eines de construcció automàtiques que faciliten la creació del codi i la detecció d'errors.

El fet d'haver realitzat un projecte similar amb anterioritat ha facilitat molt la creació d'escenes i de codi en C#. A més, durant el màster i el grau, s'adquireixen coneixements bàsics de programació en llenguatge Python, estudiant diversos conceptes informàtics com el tipus de variables, llistes, diccionaris, classes, funcions i tot tipus d'eines informàtiques que són comunes a tots els llenguatges. Per tant, no ha sigut un impacte molt gran la necessitat de programar en un llenguatge com és el C#.

5.3. Funcionament bàsic del programa

El funcionament bàsic de Unity és molt simple. Cada projecte creat es divideix en escenes. Cada escena consta de diversos "GameObjects", que és com s'anomenen tots els elements que formen l'escena, des d'un cub 3D fins a un text. A aquests GameObjects se'ls pot assignar

un Script, que és un fitxer de text amb un codi que descriurà el comportament d'aquest objecte.

Cada escena ha de constar, com a mínim, de dos elements. El primer és l'objecte càmera, que és el que s'encarrega de filmar la part de l'escena creada una vegada clicat el botó de "play" que reproduïx l'escena. Aquesta càmera pot rotar i moure's segons les necessitats del projecte. El segon element indispensable és la llum, en aquest cas, una llum direccional que representa el sol i que il·lumina completament l'escena.

Es poden visualitzar varies finestres simultàniament, però principalment la finestra on es treballa més és la de l'escena, on es poden crear, situar i modificar els diferents objectes que es desitgi introduir a l'escena. També existeix la pantalla "Game", que és on tenim una vista del que filma la càmera. A més, consta d'una finestra on es pot veure la jerarquia dels diferents elements de l'escena, i una altra anomenada "Project", on es poden organitzar els diferents arxius de l'escena. Per acabar, un element molt important de Unity és la finestra anomenada "Inspector". En aquesta finestra es poden gestionar els GameObjects, definir la seva escala, l'orientació, assignar-li un codi, una textura o un component determinat. També es pot modificar des de l'inspector, el valor de variables que dins d'un script s'han declarat com a públiques.

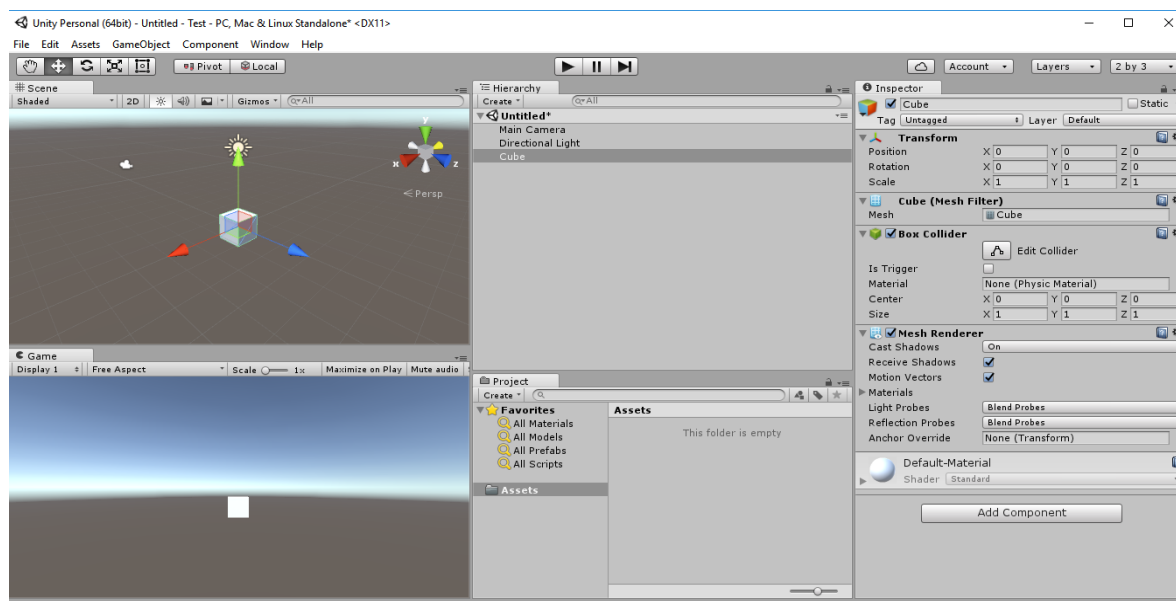


Figura 1. Interfície gràfica d'usuari del programa Unity

A la figura 1 es pot observar la interfície gràfica del software amb les diferents pantalles i elements descrits.

Com qualsevol videojoc, Unity funciona amb un "game loop", és a dir, inicialitza tots els elements, després els actualitza i finalment els renderitza (visualitza per pantalla) per a cada

frame (imatge instantània de l'escena). Per aquest motiu Unity ja té funcions predefinides, algunes de les quals apareixen al crear qualsevol script. Aquestes funcions principals són la funció Start() i la funció Update(). La primera serveix per inicialitzar variables, i la segona per definir el comportament d'actualització del GameObject.

6. Dispositius HTC

6.1. Realitat virtual

La realitat virtual es refereix a la representació d'escenes o imatges d'objectes produïts per un sistema informàtic, que dona la sensació de la seva existència real [1]. Aquest entorn virtual es visualitza normalment unes ulleres o cascs de realitat virtual. També es poden afegir altres tipus de dispositius, per interactuar amb l'entorn i millorar la sensació d'immersió. Quan l'usuari executa l'aplicació, es realitza un seguiment del seu moviment per a que coincideixi amb el moviment dins l'escena virtual. Per tant, l'usuari es pot desplaçar dins l'escena virtual. Així s'aconsegueix la immersió de l'usuari, al que li dona la sensació de que realment és dins el món virtual.

HTC és una de les empreses que ha aconseguit desenvolupar una tecnologia de qualitat per a poder visualitzar escenes de realitat virtual i interactuar amb els elements que les formen. Es considera una de les millors marques, ja que desenvolupa tecnologia puntera obtenint imatges d'alta definició i un posicionament molt precís. A la següent imatge es mostren els dispositius principals que han sortit al mercat.



Figura 2. Dispositius HTC

Al centre, a la part superior, es pot veure el HMD. Als dos costats, els dispositius amb forma cúbica són les base stations. A sota les base stations hi ha dos controladors i al centre, a la part inferior, es pot observar un tracker. En els següents apartat s'explica el principi de funcionament de cadascun d'ells.

6.2. HMD (Head Mounted Display)

Vive és un dispositiu per a visualitzar escenes de realitat virtual desenvolupat per les empreses HTC i Valve. Està dissenyat per a utilitzar-se en un espai reduït com una habitació. El dispositiu utilitza dues pantalles, una per a cada ull, amb una resolució de 1080x1200. Actualment ha sortit al mercat un nou model que permet la visualització en 4K. Aquest dispositiu pot reproduir vídeo a una freqüència de 90 fps.

Utilitza més de 70 sensors, incloent-hi un giroscopi MEMS, acceleròmetres y sensors làser. Està fet per funcionar en una àrea de seguiment limitada. Aquesta àrea és de 4,6 metres per 4,6 metres, obtenint una precisió de posicionament de menys d'un mil·límetre.

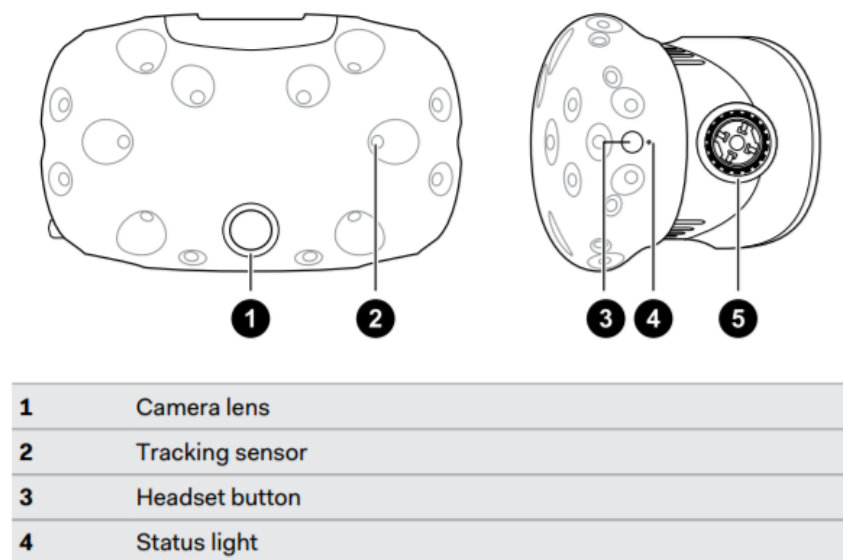


Figura 3. Head Mounted Display

El tracking es refereix al seguiment de la posició i orientació dels dispositius i és una de les funcions essencials que té el sistema de HTC Vive, per això el primer disseny que es va realitzar a la fase de desenvolupament va ser el de la posició on anirien els fotodíodes, per així tenir el millor rastreig de posicionament del casc possible. Després es va adaptar la forma del Headset a l'estructura de posicionament dels fotodíodes que ja s'havia determinat. Cada un dels fotodíodes de les ulleres va cobert per un filtre anomenat filtre IR o filtre d'infrarojos. Aquests filtres tenen la funció de reflectir i bloquejar les longituds d'ona infraroges i d'aquesta manera s'evita que apareguin interferències a l'hora de rebre les dades en el sistema. A la figura 3, es pot veure indicat amb un 2, un dels fotodíodes integrats al dispositiu. N'hi ha un per a cadascun dels orificis de la seva estructura.

6.3. Base Stations

Per a poder realitzar el tracking dels diferents dispositius és necessari utilitzar les base stations d'HTC, també anomenades lighthouses degut a la tecnologia que utilitzen. Es tracta de dos sensors encapsulats en dues caixes negres que s'encarreguen de la detecció de la posició dels diferents sensors receptors que tenen els dispositius repartits per la seva estructura.

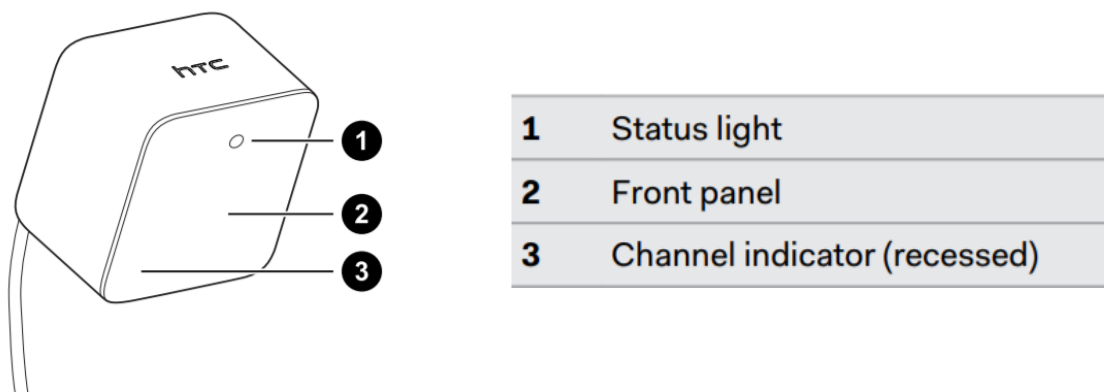


Figura 4. Base Station

Les base stations treballen amb la tecnologia Lighthouse. Aquesta tecnologia es basa en inundar l'habitació de llum no visible mitjançant làsers integrats dins les base stations per crear un camp de referència pel HMD, els trackers i els controladors. Les base stations fan la funció de far (d'aquí el nom lighthouse) i els dispositius amb els sensors que tenen repartits per tota la superfície reben el senyal, processen les dades i es posicionen a l'espai.

Les Base Stations realitzen dues accions, la primera la d'emetre feixos de llum a partir de làsers per tot el volum de joc, per tenir una referència espacial, i la segona la d'emetre flaixos periòdicament (mitjançant un conjunt de Leds) per tenir una referència temporal.

En comptes d'utilitzar una càmera per determinar on estan posicionats els objectes, utilitza llum no-visible. Un conjunt de LEDs situats dins el Lighthouse emet un flash amb una freqüència de 60 vegades per segon. Aquests flaixos serveixen de referència temporal.

El camp de posicionament es crea amb dos rotors que tenen incorporats els emissors làsers. Els dos rotors giren a 3600 rpm respecte els seus propis eixos (seixanta vegades per segon). Un rotor gira respecte l'eix vertical del sistema de coordenades de la Base Station (de baix a dalt) i l'altre respecte l'eix horitzontal (d'esquerra a dreta) i així s'abraça tot el volum d'interacció.

En la figura 5, es poden observar els dos motors que fan girar els làsers i els leds que realitzen els flaixos en la part central.

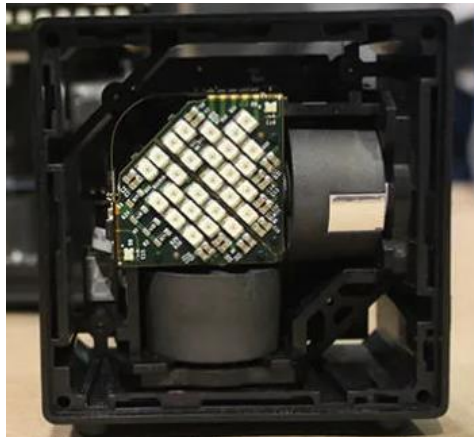


Figura 5. Tecnologia Lighthouse

La combinació d'aquestes dues tecnologies permet que els dispositius es puguin posicionar a partir dels sensors que tenen repartits per la seva estructura. Aquests sensors detecten el feix de llum, els làsers i l'instant de temps.

Quan es detecta el flaix dels leds, es comença a cronometrar el temps que passa fins que un dels làsers arriba al sensor. Aleshores, s'utilitza la relació entre la posició del sensor que ha detectat el làser i el temps que ha passat fins que l'ha detectat per a calcular la seva posició relativa respecte les estacions emissores de llum. Si el làser és detectat per suficients sensors alhora, es forma una pose, una figura 3D que permet saber amb precisió no només on està el dispositiu, sinó la orientació que té [2].

Quan s'ha realitzat aquest procés, un algoritme calcula matemàticament en quina posició exacta de l'habitació està ubicat i orientat el dispositiu respecte les Base Stations.

6.4. Controladors

Els controladors permeten generar inputs dins l'escena virtual, a través d'un pad i de diversos botons. L'usuari pot interactuar amb les escenes de realitat virtual amb aquests controladors, ja sigui per interactuar amb objectes, moure's per a l'escena o seleccionar alguna opció.

A la següent figura es mostra una representació dels controladors.

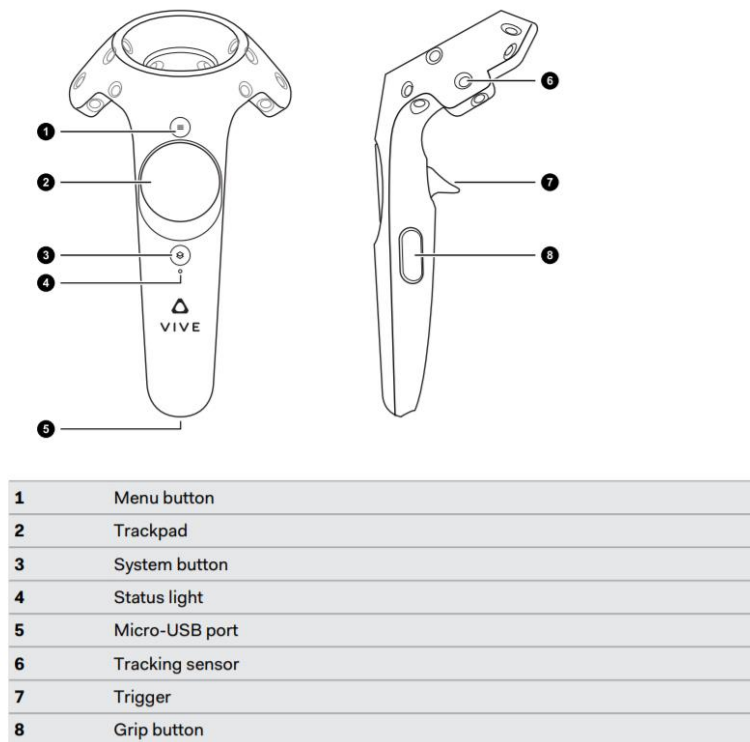


Figura 6. Controlador

Aquests controladors, com el HMD, tenen sensors basats en díodes fotovoltàics que permeten obtenir la seva traçabilitat dins de l'espai de joc. És important tenir com a mínim un controlador per a poder definir l'espai lliure del que es disposa per reproduir les aplicacions de realitat virtual. La definició de l'espai lliure s'explica amb més detall a l'apartat 7.3.3.

6.5. Trackers

Els trackers són els dispositius més recents que HTC ha tret al mercat. Les ulleres serveixen per a visualitzar l'escena i els controladors per interactuar amb ella, però no hi ha cap manera d'introduir un objecte real dins l'escena. Per això es van desenvolupar els trackers. Tenen una mida reduïda i permeten realitzar un seguiment de qualsevol objecte al que es fixin. Aquest seguiment s'anomena tracking (d'aquí prové el seu nom). El tracking es refereix al seguiment de la posició i orientació d'un objecte al llarg del temps.

Usualment s'adjunten a un objecte real mitjançant algun tipus de fixació. S'han dissenyat per a introduir objectes dins d'escenes com per exemple una pistola, una raqueta o fins i tot, amb la combinació de diferents trackers repartits pel cos, es podria fer un seguiment complet del moviment d'una persona.



Figura 7. Tracker fixat en una raqueta

Els trackers també disposen de sensors per a la seva detecció de manera precisa i es connecten mitjançant un port USB amb una connexió sense fils. Les dades que proporciona es limiten a la posició i la orientació del dispositiu.

La següent figura mostra una representació d'un tracker indicant les seves parts.

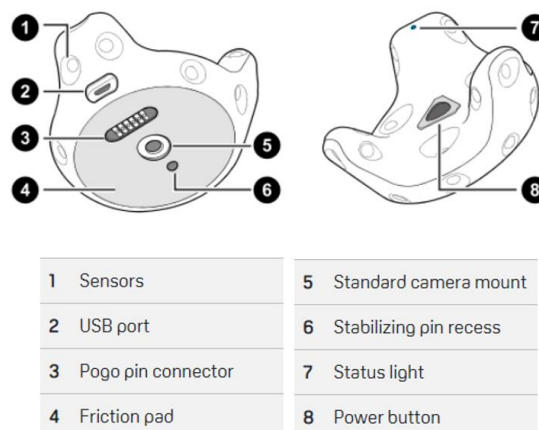


Figura 8. Tracker

6.6. Steam

Steam és una plataforma de distribució digital, gestió digital de drets, comunicacions i serveis multijugador desenvolupada per Valve Corporation, una de les dues empreses que ha desenvolupat aquests dispositius.

El software de Steam anomenat SteamVR, s'encarrega de la comunicació entre els dispositius HTC i l'ordinador. Permet definir la zona de joc, detectar els diferents dispositius i té un seguit d'opcions que permet configurar-los.

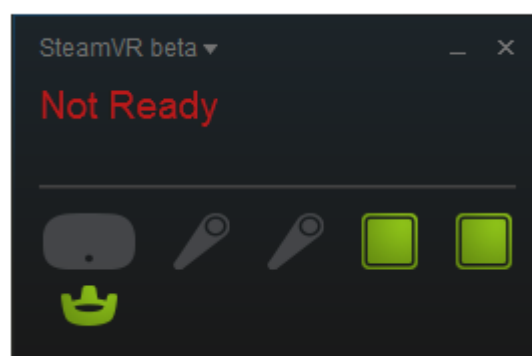


Figura 9. Software SteamVR

A la figura 9, es mostra com el software SteamVR permet detectar i configurar els diferents dispositius. Amb color verd es representen els dispositius que estan actius i en gris els que no ho estan. En aquest cas només estan connectades les dues base stations i un tracker. En canvi, el HMD i els dos controladors estan inactius.

6.7. Requeriments

El sistema d'HTC requereix d'altres prestacions pel que fa a l'ordinador que executa l'aplicació i llegeix les dades. Degut a la seva gran capacitat necessita un hardware prou potent per processar les dades a temps real i aconseguir una bona experiència per a l'usuari. Les prestacions mínimes necessàries per a utilitzar els dispositius es resumeixen en la següent taula, extreta del manual d'usuari.

Taula 1. Requeriments del computador

Component	Recommended system requirements	Minimum system requirements
Processor	Intel® Core™ i5-4590/AMD FX™ 8350 equivalent or better	Intel Core i5-4590/AMD FX 8350 equivalent or better
GPU	NVIDIA® GeForce® GTX 1060, AMD Radeon™ RX 480 equivalent or better	NVIDIA GeForce GTX 970, AMD Radeon R9 290 equivalent or better
Memory	4 GB RAM or more	4 GB RAM or more
Video output	HDMI 1.4, DisplayPort™ 1.2 or newer	HDMI 1.4, DisplayPort 1.2 or newer
USB port	1x USB 2.0 or newer	1x USB 2.0 or newer
Operating system	Windows® 7 SP1, Windows 8.1 or later, Windows 10	Windows 7 SP1, Windows 8.1 or later, Windows 10

Com es pot observar a la taula, és molt important la capacitat de processament per executar les aplicacions de realitat virtual, es requereix d'un processador i una targeta gràfica potents i a més, cal disposar de port USB 2.0.

Durant el projecte, es pretenia utilitzar l'ordinador personal per a desenvolupar les escenes per una raó de practicitat, però aquest equip no complia amb els requeriments, sobretot pel que fa a la capacitat de la targeta gràfica (GPU). És per això que es va facilitar un ordinador al Centre de Realitat Virtual (CRV) a la Facultat de Matemàtiques i Estadística compatible amb el projecte.

Les característiques d'aquest ordinador es resumeixen en la taula següent.

Taula 2. Característiques del computador utilitzat

Processador	Intel (R) Core (TMS) i5-4460 3.2 GHz
GPU	NVIDIA GTX 1050
Memòria RAM	8 GB
Sistema Operatiu	Windows 10 Pro

Aquest ordinador és de la marca LG i com es veu a la taula, disposa d'una GPU dedicada suficientment potent per a poder utilitzar els dispositius de realitat virtual.

7. Implementació d'HTC Vive amb Unity

7.1. Introducció: aplicacions dels dispositius i antecedents

Existeixen una gran quantitat d'aplicacions per a reproduir amb el headset (ulleres o casc) de realitat virtual d'HTC i totes elles estan pensades per a ser executades amb un ordinador en un espai reduït com una habitació. Hi ha aplicacions de tot tipus, que ofereixen una immersió molt realista. Comercialment, aquesta tecnologia s'ha pensat especialment pel món dels videojocs - un mercat amb milions d'usuaris que permet la viabilitat del producte - tot i que encara l'alt preu d'adquisició dels dispositius de realitat virtual no ha deixat que es popularitzi de forma massiva.



Figura 10. Aplicació de realitat virtual

A part dels videojocs, també s'han desenvolupat altres tipus d'aplicacions relacionades amb diversos camps, com per exemple l'aplicació de Google Earth VR. Aquesta permet visualitzar una gran quantitat de llocs geogràfics de manera virtual, gràcies a la gran quantitat de mapes 3D que ha generat la companyia. L'usuari pot decidir com moure's per qualsevol zona mitjançant els controladors com es mostra a la figura 11.

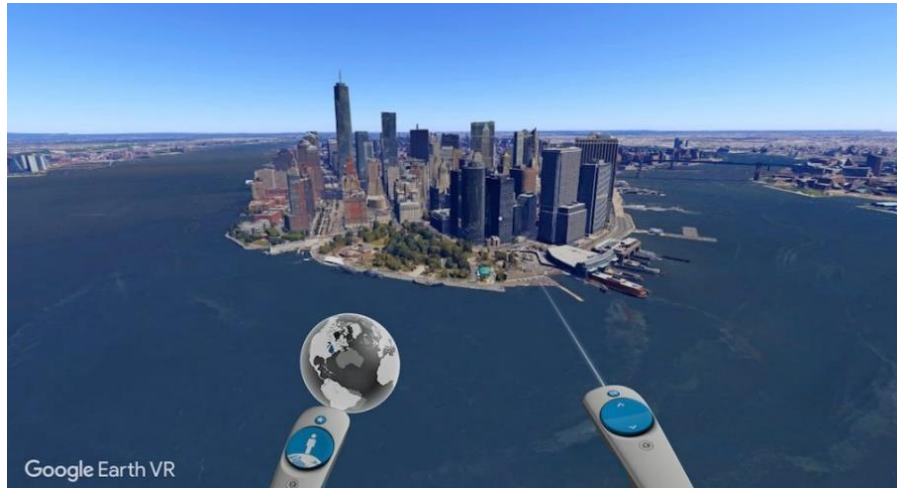


Figura 11. Aplicació Google Earth VR

Es preveu que en un futur proper, el preu d'aquests dispositius tendirà a reduir-se fins a guanyar pes en el mercat, permetent que qualsevol persona pugui accedir a aquesta tecnologia.

És per això, que és interessant per al projecte, analitzar les possibilitats que ofereix aquesta nova tecnologia. En concret, el Centre de Realitat Virtual disposa dels dispositius d'HTC Vive, que són probablement els més avançats en aquest sector.

La intenció és definir la metodologia i el desenvolupament dels recursos necessaris per a crear aplicacions de realitat virtual i augmentada utilitzant principalment els trackers d'HTC per al posicionament i tracking d'objectes virtuals.

7.2. Plugin per Unity: SteamVR

Com a motor de jocs, Unity té l'interès de facilitar als seus desenvolupadors la possibilitat de crear aplicacions per al màxim tipus de plataformes. És per això, que per a la majoria d'elles s'ha creat un plugin que permet accedir de manera fàcil a totes les funcions de la plataforma que es desitgi. Un plugin és aquella aplicació que, en un programa informàtic, afegeix funcionalitats addicionals o una nova característica de software.

Per a poder realitzar aplicacions utilitzant els dispositius HTC Vive amb Unity, és necessari descarregar el plugin de SteamVR. Aquest plugin conté una sèrie de codis i altres elements necessaris per al correcte funcionament dels dispositius i permet gestionar-los des de Unity. Aquest plugin es pot obtenir de forma totalment gratuïta des de la "Asset Store", que és la plataforma oficial de Unity per descarregar assets per al motor de jocs. En aquesta plataforma es poden obtenir tot tipus d'assets, que són arxius relacionats amb Unity: des de codi en C# fins a models 3D o animacions.

A la figura 12, es pot veure el plugin que s'ha descarregat per al projecte.

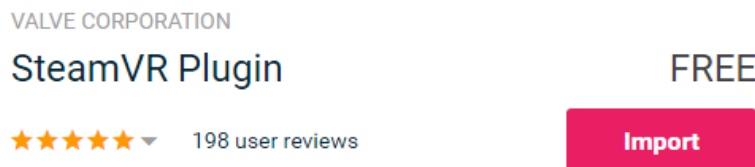


Figura 12. Plugin SteamVR

Una vegada descarregat, cal importar-lo al projecte de Unity, d'aquesta manera quedaran els arxius guardats a la carpeta del projecte dins una carpeta anomenada SteamVR i ja estaran llestos per a fer-los servir.

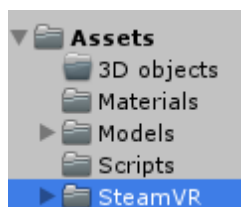


Figura 13. Fitxers del plugin SteamVR

7.3. Desconnexió del casc

7.3.1. Motivació

Actualment, els dispositius de realitat virtual tenen un preu molt elevat per a l'usuari mig. És cert que aquest any 2018, han sortit al mercat molts headsets diferents on s'introdueix el smartphone per reproduir escenes de realitat virtual. Aquests nous dispositius tenen un preu molt més assequible, però no arriben al nivell de qualitat dels dispositius d'HTC. Tenen una precisió inferior i la qualitat de la imatge encara ha de millorar molt. Tot i així, és interessant realitzar una combinació de la tecnologia més avançada d'HTC i els smartphones.

Un kit d'HTC Vive costa actualment al voltant d'uns 500 euros. Aquest kit conté el headset, dos controladors i dues base stations. El casc de visualització (HMD) és el que suposa un cost més elevat en proporció a tot l'equip. Per altra banda els trackers són els dispositius més econòmics de la marca, amb un cost al voltant d'uns 100 euros.

En aquest projecte es pretén realitzar aplicacions que necessitin del mínim de dispositius amb el mínim cost per a executar-les. És per això, que s'ha analitzat la opció de no utilitzar el casc per a visualitzar les escenes virtuals i s'han escollit alternatives més econòmiques. Per aquest motiu, les aplicacions desenvolupades es basen en el posicionament i tracking mitjançant els trackers i la visualització de les escenes amb dispositius mòbils com els smartphones.

7.3.2. Procediment

Abans de començar amb la desactivació del casc, cal descarregar el següent Software:

- Software SteamVR versió Beta
- Python 3.6 (Anaconda Package)
- Pyopenvr (instal·lar amb la comanda: `pip install pyopenvr`)
- Triad_openvr

Per tal d'obtenir les dades del tracker sense haver de connectar el casc, és necessari realitzar una sèrie de passos [3]. El software que gestiona aquests dispositius és el SteamVR, per tant, cal modificar el fitxers de configuració que obliguen a connectar el casc per al correcte funcionament dels dispositius. Concretament s'han de modificar dos arxius de configuració.

El primer està localitzat a la carpeta següent:

Steam\steamapps\common\SteamVR\drivers\null\resources\settings\default

Cal canviar la opció enable a "true".

```
"driver_null" : {  
  "enable" : true  
  "serialNumber" : "Null Serial Number",  
  "modelName" : "Null Model Number",  
}
```

Figura 14. Primera modificació

El segon arxiu que cal modificar està situat al directori següent:

Steam\steamapps\common\SteamVR\resources\settings\default

S'han de modificar les variables requiredHMD a "false", forcedDriver a "null" i activateMultipleDrivers a "true".

```
{  
  "steamvr": {  
    "requireHmd": false,  
    "forcedDriver": null,  
    "activateMultipleDrivers": true,  
  },  
}
```

Figura 15. Segona modificació

D'aquesta manera es modifiquen els arxius de SteamVR que gestionen els dispositius, habilitant l'opció d'utilitzar el tracker i els controladors sense necessitat de tenir el HMD. Una vegada modificats els arxius, al obrir el programa SteamVR, ja no requerirà el casc per a poder funcionar correctament.

7.3.3. Zona de joc i assignació dels índexs

Per limitar la zona on es mou l'usuari, s'han de definir els límits de l'habitació o sala on es vulgui executar les aplicacions. Aquesta zona s'anomena Play Area (àrea de joc) i el plugin de Unity les representa amb un rectangle de color blau com es mostra a la figura 16.

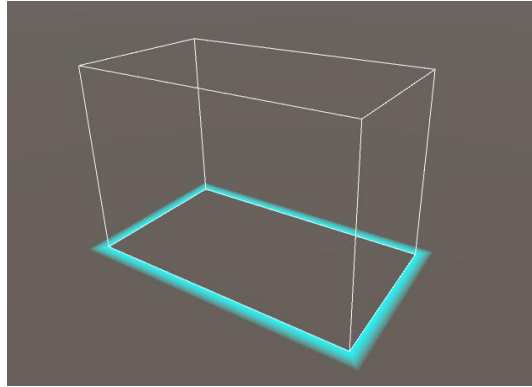


Figura 16. Play Area

Aquesta zona s'ha de definir mitjançant el software SteamVR. Normalment, per a fer-ho sol·licita la connexió del casc i dels controladors, però al haver desconnectat el casc la metodologia canvia. Al no ser una manera convencional de definir aquesta zona, no hi ha documentació al respecte i per això s'han realitzat proves pràctiques per entendre el seu funcionament.

Quan es desconnecta el casc, ja no el sol·licita, però s'ha arribat a la conclusió de que com a mínim, han d'estar connectats una base station, un controlador i un tracker per a poder definir-la. Els passos a seguir si comptem amb aquests dispositius mínims, sí són els habituals. Primer, s'han d'activar els dispositius per a poder detectar-los, quan el SteamVR ja els ha reconegut s'ha d'anar a l'opció de definir habitació que facilita aquest software. A continuació es demana que s'esculli entre dues opcions per a definir l'habitació. En la primera opció, l'usuari es pot moure lliurement per a l'escena. En cas contrari, cal seleccionar l'opció en que l'usuari es mantindrà fix en un punt. Després cal agafar el controlador i apuntar al monitor de l'ordinador mentre es prem el botó del controlador. Per calibrar l'alçada de la base station respecte el terra, s'ha de col·locar un dispositiu al terra i prémer continuar. Finalment amb el controlador, es seleccionen prement el botó, els punts que defineixen el perímetre de la zona.

A la següent figura es mostra com s'ha definit l'àrea de joc en el Centre de Realitat Virtual durant el projecte.



Figura 17. Play Area definida per al projecte

És de vital importància entendre els índexs que assigna als diferents dispositius ja que quan es vulgui obtenir les dades de cadascun, s'haurà de seleccionar l'índex del dispositiu en qüestió. De manera empírica s'ha vist, que al no haver-hi un casc connectat, el dispositiu que rep el primer índex és el primer en ser detectat pel software SteamVR. Els següents índexs (2 i 3) s'assignen automàticament a les base stations. A partir d'aquí, a mesura que es van detectant nous dispositius, se'ls assigna els índexs en ordre de detecció.

7.3.4. Creació d'una escena

Quan ja s'ha definit la Play Area, després d'haver modificat els arxius per a desconectar el casc, s'ha de crear una escena nova i importar el plugin de SteamVR.

Per iniciar una escena, s'ha d'assignar el script del plugin anomenat **SteamVR_Camera** a la càmera principal de l'escena. Aquest acció afegeix dos scripts nous a la càmera. Cal anar a la finestra inspector per prémer la opció del nou script afegit anomenada Expand [4]. Al fer això apareixen nous components com es mostra a la figura 18.

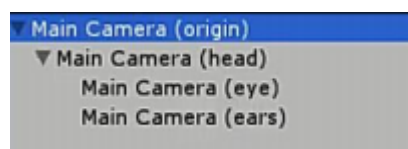


Figura 18. Jerarquia dels components de l'escena

S'ha eliminat el component **Main Camera (ears)** ja que no interessa gestionar el so en aquesta escena i s'ha eliminat també l'objecte Main Camera origin (sense eliminar els altres elements). Finalment s'ha anomenat al Main Camera (head) com a "Tracker", ja que representarà la posició i orientació del tracker.

Per obtenir les dades de la posició i l'orientació dels diferents dispositius cal assignar un script als objectes. Aquest script s'anomena **SteamVR_TrackedObject** i obté les dades de la posició i l'orientació del dispositiu amb l'índex que es seleccioni. Al assignar les dades del tracker a la càmera, s'aconsegueix que la càmera segueixi la mateixa trajectòria que el tracker, fent possible el moviment de l'usuari dins l'escena virtual.

L'elecció dels dispositius es gestiona a partir d'índexs. L'assignació dels índexs ja s'ha explicat en aquest document en el punt 7.3.3.

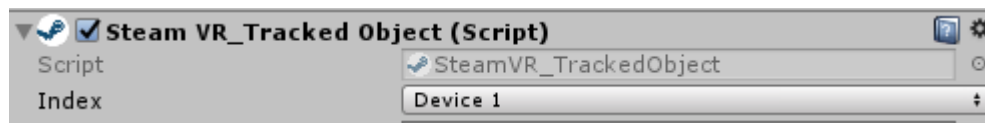


Figura 19. Índexs dels dispositius

A l'objecte anomenat "Tracker" que conté la càmera se li ha assignat l'índex 1.

S'han afegit també dos elements que representen els lighthouses. A aquests dos GameObjects se'ls ha assignat el script **SteamVR_TrackedObject** seleccionant respectivament els índexs 2 i 3 que representen els lighthouses.

És molt important, ja que no es farà servir el casc per visualitzar l'escena sinó una pantalla, que a la càmera que es fa servir anomenada Main Camera (eye) es seleccioni la opció de Target eye a "None" com es veu a la figura 20. Això farà que simplement es visualitzi l'escena amb la càmera en comptes d'intentar canviar la resolució per adaptar-la a les possibles ulleres de realitat virtual que es puguin connectar. A més, així també evitem que modifiqui el camp de visió de la càmera i que es creï un efecte borrós perjudicant la qualitat de l'escena.

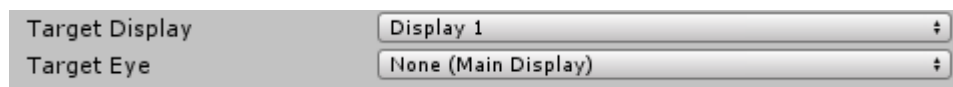


Figura 20. Configuració del target Eye

Finalment s'ha afegit a l'escena el prefab anomenat **CameraRig** de la carpeta Prefabs del plugin. El CameraRig representa la zona accessible o zona de joc. La zona es pot escollir manualment o es pot seleccionar la que s'ha calibrat per a una habitació concreta. Per evitar posicionar objectes a llocs inaccessibles o evitar possibles impactes amb objectes al fer aplicacions de realitat virtual, s'ha decidit utilitzar l'opció de zona calibrada com es mostra a la figura 21.

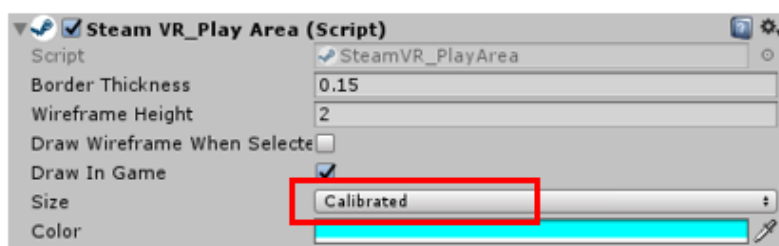


Figura 21. Selecció de la Play Area calibrada

S'han eliminat els components dels controladors i la càmera que apareixen per defecte ja que no són necessaris per aquest projecte. La zona de joc es fa servir de referència en sí, és a dir, per a que es situïn els objectes on realment estan respecte la habitació real, cal utilitzar la zona calibrada com a origen. És per això que s'han definit els dispositius com a fills del CameraRig quedant l'estructura de l'escena de la següent manera:

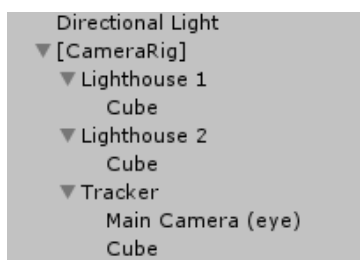


Figura 22. Jerarquia de l'escena

Com es pot observar a la figura 22, s'han assignat cubs de colors als diferents dispositius per a poder visualitzar la seva posició i orientació per pantalla.

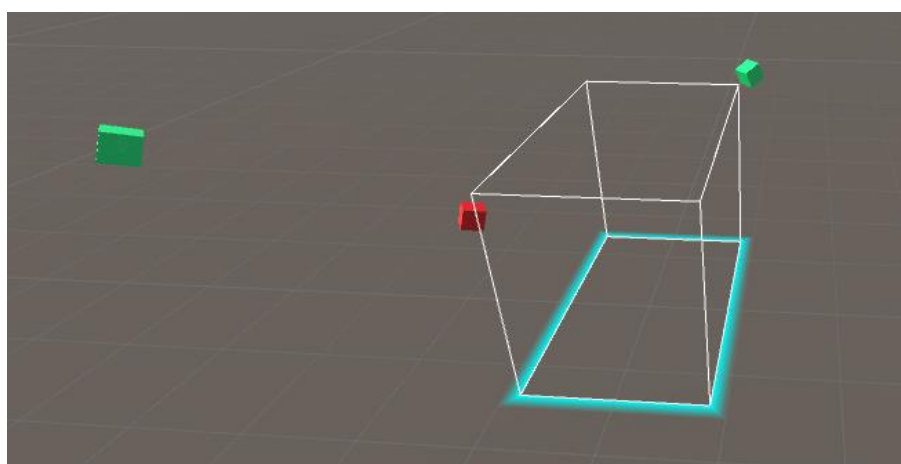


Figura 23. Representació de les posicions dels dispositius

Els cubs verds de la figura 23, representen les base stations i el cub vermell, representa el tracker.

Aquesta primera escena, serveix com a “demo” per a la gestió del plugin i permet entendre de manera fàcil com s'obtenen les dades dels diferents dispositius. Una vegada creada ja es poden obtenir les dades de posició i orientació de qualsevol dispositiu HTC i utilitzar-les per a crear qualsevol tipus d'aplicació.

Aquesta aplicació es pot fer servir en un cas d'ús real al Centre de Realitat Virtual. Es tracta de la Powerwall, una pantalla d'enorme resolució i grans dimensions que serveix per a visualitzar aplicacions de realitat virtual. Normalment s'escull una persona com a gestora de l'aplicació, encarregada d'anar seleccionant el que es desitgi visualitzar. Els altres usuaris, només s'han de situar davant la gran pantalla per a visualitzar l'escena amb unes ulleres especials que creen l'efecte de profunditat. D'aquesta manera, els usuaris tenen la sensació que el que s'està visualitzant, sobresurt de la pantalla creant un efecte 3D completament immersiu.

Al centre de realitat virtual disposen d'aquesta tecnologia, però la gran problemàtica és la infraestructura que s'ha d'instal·lar per a la detecció de l'usuari gestor de l'aplicació. L'estructura es basa en dues barres metàl·liques verticals que suporten una estructura horitzontal situada a més de dos metres d'altura, on s'integren els diferents sensors que realitzen el posicionament i el tracking de la persona que portarà una sèrie de receptors.

El fet d'introduir la tecnologia d'HTC a la Powerwall, permet tenir una instal·lació menys intrusiva i fàcil de transportar. La idea és posicionar a la persona amb un tracker i que aquesta el vagi movent per l'escena per visualitzar el que desitgi.



Figura 24. Powerwall

7.4. Comunicacions entre dispositius

7.4.1. Limitacions dels dispositius mòbils

Les prestacions dels dispositius mòbils han evolucionat molt els últims anys, tant pel que fa a la capacitat de processament com de memòria entre d'altres característiques. Tot i així, la realitat virtual és un camp molt exigent pel que fa a la capacitat de l'equip que l'executa, especialment quant es tracta del processament gràfic. Les targetes gràfiques dels dispositius mòbils no són suficientment potents per a poder gestionar una aplicació amb dispositius HTC. A més, per a obtenir les dades de tracking cal estar connectat al software de SteamVR com ja s'ha comentat anteriorment i per tant, només amb un dispositiu mòbil no podríem executar una aplicació com la que es vol desenvolupar.

7.4.2. Wifi / Bluetooth

És per això, que és necessari buscar una alternativa a aquest problema. La solució al problema és transferir les dades de la posició i orientació dels dispositius des de l'ordinador cap al dispositiu, de tal manera que l'aplicació llegeixi les posicions i orientacions dels dispositius HTC sense necessitat de connectar-se directament a ells.

La idea és poder executar l'aplicació a l'ordinador i al dispositiu mòbil simultàniament i transferir les dades a temps real des de l'ordinador al dispositiu. Per fer això, és necessària una connexió sense fils, ja que haver de connectar-se físicament amb cables reduiria molt el

moviment de l'usuari i afectaria a l'experiència de l'aplicació.

Els dos protocols més utilitzats per a transferir dades sense fils actualment són el Bluetooth i el Wifi. Per tant, per poder transferir dades entre el PC i el dispositiu, caldrà implementar una d'aquestes dues opcions. Per a poder escollir quin tipus de connexió s'adequa millor a la situació, cal destacar les avantatges i inconvenients de cadascun.

El Bluetooth és un protocol molt utilitzat en smartphones i petits gadgets que es connecten a altres dispositius com per exemple auriculars, altaveus sense fils o el mans lliures per al cotxe. El principal avantatge que té, és que no necessita que els dispositius que es comuniquen estiguin connectats a la mateixa xarxa. Per altra banda, la transferència de dades es realitza amb unes velocitats significativament inferiors al Wifi, fet que pot tenir un efecte molt negatiu en una aplicació que transfereix dades a temps real causant retards.

Per altra banda, el Wifi permet crear xarxes més grans i transferir dades a velocitats superiors entre dispositius. L'inconvenient principal és que els dos han d'estar connectats a internet i a la mateixa xarxa. En aquest cas, això no és un problema pel fet de que l'aplicació s'executarà en un ordinador que sempre disposarà d'internet i la gran majoria de dispositius mòbils també incorporen la possibilitat de connexió Wifi [5].

7.5. Servidor / Client

Unity és un motor de jocs, i com principalment s'utilitza per al desenvolupament d'aquests, incorpora la possibilitat de crear jocs on els usuaris es connectin entre ells per compartir dades o competir uns amb altres. Concretament, facilita la llibreria "UnityEngine.Networking" que conté un seguit de mètodes, classes i funcions per implementar servidors i clients.

7.5.1. Funcionament

Per a la transferència de dades entre l'ordinador i el smartphone, s'ha definit un model Client/Servidor.

Un servidor és un ordinador o un altre tipus d'equip informàtic encarregat de subministrar la informació a una sèrie de clients. La informació que pot transmetre és múltiple i variada: des d'arxius de text, imatges o vídeos. En aquest cas el servidor és l'ordinador, i les dades que

transmet són la posició i la orientació dels dispositius que s'utilitzin.

El client es defineix com una aplicació informàtica o un ordinador que consumeix un servei de forma remota en un altre ordinador conegut com a servidor. En aquest cas, el client és el dispositiu mòbil que executa la aplicació i llegeix les dades de posició i orientació enviades pel servidor.

Per aconseguir una transferència sense fils, s'ha implementat una connexió mitjançant Wifi, ja que com s'ha comentat en l'apartat 7.4, és la més indicada pel cas d'ús.

A continuació es mostra un esquema del model Client/Servidor:

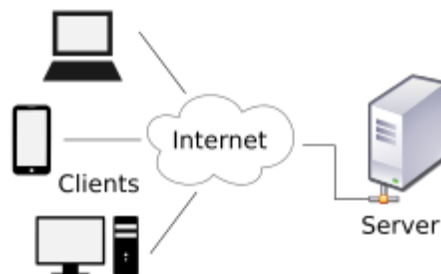


Figura 25. Esquema Servidor/Client

La metodologia és la següent: el client sol·licita la informació al servidor i aquest respon amb les dades sol·licitades. Això s'estableix a partir d'un canal i un port, que s'han de definir prèviament. Per a aquest projecte s'ha utilitzat el canal 888 per a la transferència de dades i el port 25000.

A més, per a connectar-se al servidor, el client ha d'introduir la IP del servidor en qüestió, per a que la aplicació conegui quin és el servidor del qual ha de llegir les dades. Per tant, l'usuari de l'aplicació haurà d'introduir-la.

7.5.2. Implementació

Per a la transmissió de dades s'ha ideat un estructura dels elements de l'escena concreta, que facilita molt el desenvolupament d'aplicacions. Es tracta de crear una mateixa escena amb un servidor i un client. Quan es compili l'aplicació per als dispositius Android, s'ha de desactivar el servidor i deixar activat el client, ja que aquesta aplicació s'instal·larà al

smartphone i serà la que actui com a client.

Per altra banda, quan s'executi l'aplicació des del PC per obtenir les dades dels dispositius i enviar-les als clients, s'haurà de desactivar el client i activar el servidor. A la següent imatge es pot veure com l'escena està preparada per a que l'executin com a servidor, ja que el client està en un to gris més clar que indica que no és actiu.

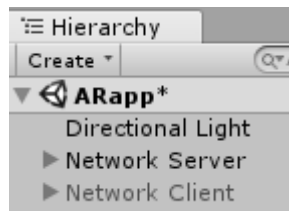


Figura 26. Estructura de l'escena Servidor/Client

S'ha desenvolupat un script en llenguatge C# per al servidor i un altre per al client. El script del servidor s'encarrega de crear el servidor i guardar en variables les dades obtingudes dels trackers. Després, crida a la funció "Listen" que escolta les peticions que es realitzen a través d'un port definit prèviament. Quan rep una sol·licitud, crea un missatge d'una classe definida específicament per enviar les dades. Aquestes dades s'envien a través d'un canal, que com ja s'ha comentat amb anterioritat és el 888. Com a informació complementaria, aquest script mostra per pantalla la IP que se li ha assignat a l'ordinador, per a que l'usuari del smartphone la pugui conèixer i l'introdueixi a l'aplicació per establir la comunicació.

El script del client per altra banda, s'encarrega de crear un client. Aquest client per a cada bucle, envia una sol·licitud al servidor, mitjançant el port, el canal i la IP definides. A continuació llegeix el missatge i guarda les dades obtingudes en variables.

Com s'ha esmentat en l'apartat anterior, la informació que ha de transmetre el servidor són la orientació i la posició dels dispositius. Aquesta informació, a Unity, és defineix amb dues classes específiques. La posició és de la classe "Vector3", que com el seu nom indica és un vector de tres components. Cada component del vector representa respectivament les coordenades x,y i z en l'espai tridimensional. Per altra banda, la orientació és de la classe "Quaternion". Un quaternió és una forma de representar una rotació. És un vector de quatre elements com el de la equació 1.

$$q = \left(\cos\left(\frac{\theta}{2}\right), u \cdot \sin\left(\frac{\theta}{2}\right), v \cdot \sin\left(\frac{\theta}{2}\right), w \cdot \sin\left(\frac{\theta}{2}\right) \right) \quad (\text{Eq. 1})$$

El quaternions representen una rotació respecte un sol eix. El vector de quatre components de la equació 1, representa el vector (u,v,w) i l'angle de gir θ .

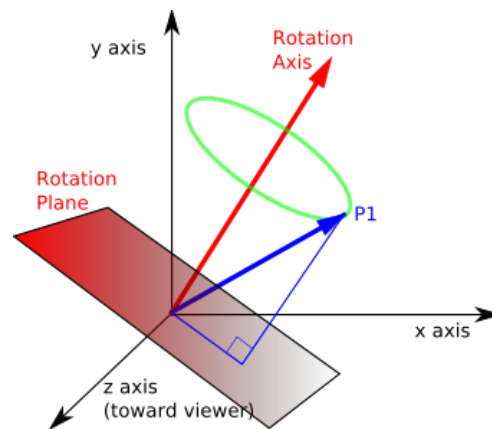


Figura 27. Representació d'un quaternió

En aquest camp i en altres com la robòtica, s'utilitza sovint aquest tipus de rotacions en comptes dels angles d'Euler per a evitar un problema anomenat gimbal lock. El gimbal lock és la pèrdua d'un grau de llibertat al realitzar certes rotacions en un ordre que ocasiona la superposició de dos eixos de rotació. Això pot portar molts problemes, com obtenir configuracions no desitjades d'un model o inclòs orientacions errònies.

Per tant, s'ha definit una classe derivada de la classe missatge amb els atributs següent:

```
//Creem la classe per definir els atributs que volem enviar
public class CameraMessage : MessageBase
{
    //Decidim els atributs que tindrà la classe creada
    public Quaternion tracker_orientation;
    public Vector3 tracker_position;
}
```

Figura 28. Implementació de la classe missatge

Aquesta classe permet enviar posicions amb el format Vector3 i rotacions representades per quaternions i ha d'estar definida tant al script del client com al del servidor per a que sigui reconeguda.

Amb aquests codis creats, ja es pot obtenir les dades de tracking i enviar-les al dispositiu mòbil

per a que les utilitzi. Amb el mode Client/Servidor implementat, ja es pot començar a desenvolupar aplicacions de realitat virtual.

A continuació es mostra un esquema de les comunicacions entre dispositius d'aquest projecte.



Figura 29. Esquema de la comunicació entre dispositius

7.6. Aplicacions Android

7.6.1. Com crear una app per Android

Un dels motius per als que s'ha escollit el motor de jocs Unity per a la realització del projecte és la seva versatilitat pel que fa a les plataformes amb les que pot treballar. Unity permet desenvolupar qualsevol aplicació i compilar-la després en una gran quantitat de plataformes. Aquest fet és de gran utilitat pel projecte, ja que té compatibilitat amb els dispositius mòbils, tant en els de sistema operatiu iOS com els d'Android.

Els dispositius iOS tenen un sistema més restrictiu i tancat. Per a poder desenvolupar una aplicació i descarregar-la en un dispositiu mòbil amb aquest sistema operatiu, és necessari donar-se d'alta com a desenvolupador d'aplicacions per iOS. El fet de ser desenvolupador iOS comporta pagar una quota de 100 dòlars anuals i les aplicacions només poden ser descarregades des de la botiga d'aplicacions oficial de l'empresa Apple anomenada AppleStore. L'aplicació ha de ser validada i acceptada pel sistema per a que es pugui publicar, havent de complir amb tota la normativa i les guies per a desenvolupar aplicacions.

Per altra banda, els dispositius Android tenen un sistema més obert, amb més facilitats per a desenvolupadors d'aplicacions. Si es vol crear una aplicació, per a que aparegui a la botiga oficial anomenada PlayStore, també és necessari estar donat d'alta com a desenvolupador

Android i pagar una quota única de 25 dòlars. El gran avantatge és que en sistemes Android, sí és possible descarregar aplicacions de llocs aliens a la botiga oficial, permetent provar-les sense necessitat de suposar un cost. Per tant, aquesta és la millor opció per a un projecte de recerca com és el cas.

Les aplicacions Android són fitxers apk, que només cal copiar en algun directori del dispositiu i prémer sobre el fitxer per a instal·lar-lo. Per a crear el fitxer apk amb Unity, cal canviar la plataforma de treball a Android tal i com es mostra en la figura següent.

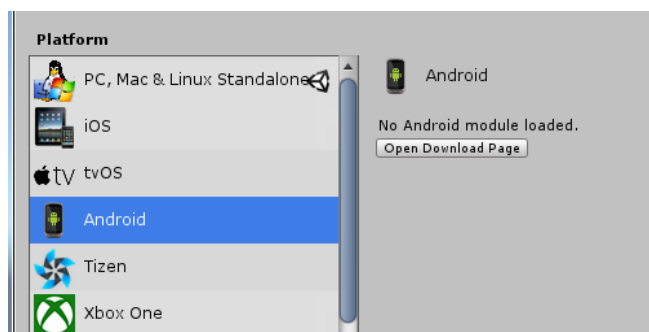


Figura 30. Opció de la plataforma Android

7.6.2. Requeriments Android

Quan es vol compilar l'aplicació, Unity demana la ruta de dues carpetes. La ruta dels directoris s'ha d'introduir a l'apartat de preferències de Unity. La primera ruta conté el SDK (Software Development Kit) de Android per a poder crear l'arxiu amb format apk. A més, també demana la ruta del JDK (Java Development Kit) [6].

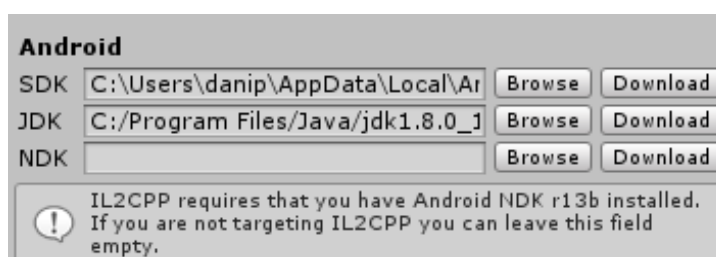


Figura 31. Ruta del SDK i el JDK

Per últim, és necessari introduir algunes dades pel que respecta a l'aplicació. En l'apartat de Player Settings, abans de compilar l'aplicació, apareixen diverses opcions a la finestra de l'inspector. En aquesta finestra es pot introduir la imatge del logo de l'aplicació per exemple. Però les dades que sí han d'aparèixer obligatòriament per a crear l'arxiu apk són: el nom de l'empresa que desenvolupa el producte i el nom de l'aplicació. També cal introduir la versió

d'Android mínima que ha de tenir el dispositiu per a poder reproduir-la. En aquest cas s'ha escollit la "4.1 Jelly Bean", que es la versió més antiga. S'ha escollit per a que la pugui executar el màxim nombre de dispositius.

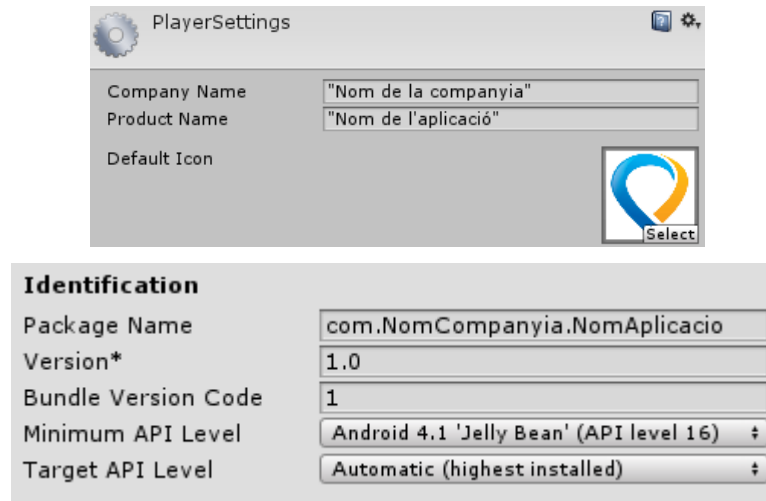


Figura 32. Nom de la companyia i de l'aplicació

Una vegada creat l'arxiu apk que conté l'aplicació, cal fer alguns petits canvis a la configuració del dispositiu, ja que es considera que descarregar aplicacions de llocs desconeguts pot suposar un perill per a la seguretat de l'usuari i de les seves dades.

Primerament, s'ha d'activar el mode de desenvolupador d'Android. Això es pot fer clicant a la informació del dispositiu a ajustos. A l'apartat informació del dispositiu cal clicar 7 vegades el número de compilació [7].

Després de clicar 7 vegades, apareixen noves opcions al dispositiu pensades per a facilitar la creació d'aplicacions als desenvolupadors. Per a poder descarregar aplicacions de fonts externes a la PlayStore cal activar l'opció de depuració USB, situada a les noves opcions que s'han activat. Per últim, per defecte no es permet descarregar aplicacions externes a no ser que l'usuari ho desitgi deliberadament. És per això que també existeix la opció de descarregar aplicacions de fonts externes i s'ha d'activar.

Una vegada ja s'ha configurat el dispositiu i s'ha canviat a la plataforma Android, només cal compilar l'aplicació per a poder provar-la, copiant-la en una carpeta del dispositiu i descarregant-la. Es poden compilar totes les escenes que es desitgin i apareixen en l'ordre que s'han afegit.

7.7. UI: Interfície de l'usuari

La interfície de l'usuari permet que aquest interaccioni amb l'aplicació. Usualment conté una sèrie de botons o menús que permeten seleccionar les diferents opcions de les que disposa l'aplicació. En aquest cas, com l'usuari l'executarà en un dispositiu tàctil, ja sigui un smartphone o una tablet, els elements de la interfície són botons tàctils.

El fet de disposar d'una interfície per a l'usuari, facilita la introducció de dades i la personalització de l'aplicació. En aquest cas, per generalitzar l'ús de la aplicació en qualsevol sala o habitació que disposi dels elements necessaris, s'ha creat un menú de selecció de la IP. L'usuari s'encarrega d'introduir la IP assignada al PC que estigui executant el codi del servidor i d'aquesta manera l'aplicació es connecta amb èxit a qualsevol ordinador. Per minimitzar el temps de proves durant el projecte, s'ha creat un botó que assigna directament la IP de l'ordinador que s'ha fet servir per al desenvolupament del projecte.

Com es pot veure a la figura 33, aquest botó conté un text que indica "Connect CRV", que és el Centre de Realitat Virtual de la Facultat de Matemàtiques i Estadística on s'ha realitzat el projecte. L'ordinador utilitzat està connectat amb un cable ethernet a la xarxa, i sempre té assignada la mateixa IP: 147.83.172.60.

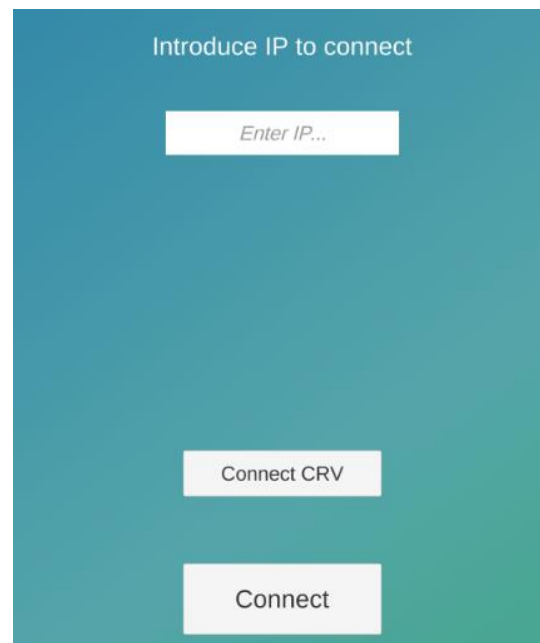


Figura 33. Menú de selecció de la IP

Si s'introdueix la IP de forma manual, després d'introduir tots els dígit, s'ha de prémer el botó continuar. Després es carrega una nova escena on es visualitza la IP introduïda i ofereix la opció de tornar a introduir-la si hi ha hagut algun error (figura 34). En cas de que sigui correcte, s'ha de prémer el botó de continuar i es carrega l'escena principal de l'aplicació.

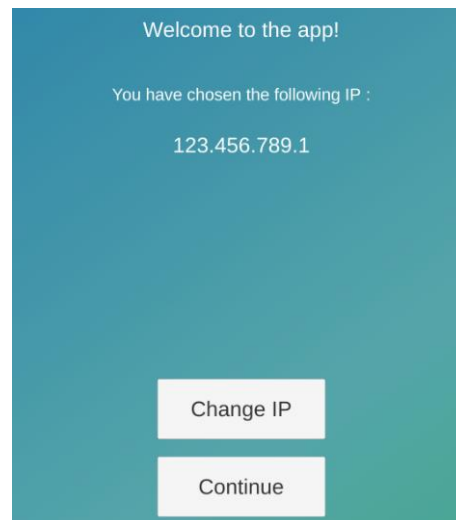


Figura 34. Menú de canvi de IP

Unity facilita la possibilitat de crear interfícies d'usuari (UI). Per a crear-ne una, és necessari introduir a l'escena un GameObject anomenat Canvas. El Canvas representa la zona on es visualitzen els elements de la UI. Per defecte es situa per sobre la pantalla i és independent al moviment de les càmeres o dels objectes dins l'aplicació. Simplement es superposa en una posició fixa.

Al crear un Canvas, automàticament apareix l'element "EventSystem". Aquest element permet enregistrar els esdeveniments que succeeixin relacionats amb la interfície de l'usuari, ja sigui el fet de prémer un botó, fer un gest sobre la pantalla o introduir dades amb el teclat virtual.

Al Canvas es poden introduir diversos elements, com per exemple text, botons, camps d'input per a introduir dades o elements per realitzar scrolls de la pantalla.

A la finestra de l'inspector es pot decidir on es situen els objectes en funció de la pantalla. Per exemple, si es vol situar en l'extrem inferior esquerre, es pot seleccionar a quina distància es vol posar respecte aquest vèrtex. Així, si s'executa l'aplicació en pantalles de diferents mides, el botó sempre es situarà al mateix lloc. A la següent figura es poden veure totes les posicions del Canvas en les que es poden introduir els elements que el formen.

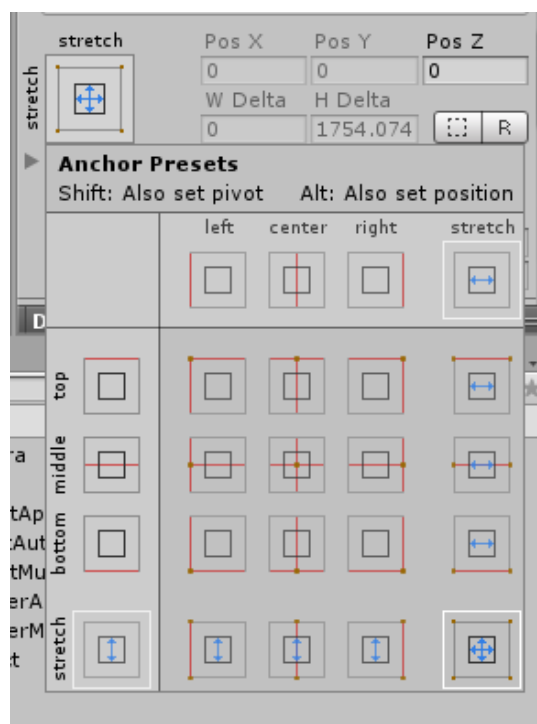


Figura 35. Posicionament dins el Canvas

Finalment s'ha creat un script anomenat "Ulmanager" que s'encarrega de gestionar el comportament de la UI. Aquest codi defineix quina acció es realitza al prémer cada botó i guarda les dades introduïdes per l'usuari en variables per a poder utilitzar-les posteriorment. Quan es crea l'aplicació per Android, s'han d'afegir les escenes que es vulgui utilitzar a l'aplicació com es mostra a la figura 36.

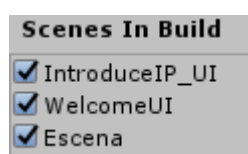


Figura 36. Selecció d'escenes a compilar

Per a totes les aplicacions s'han utilitzat les dues escenes ja comentades per a que l'usuari introdueixi la IP i s'estableixi la connexió. A més, s'afegeix l'escena principal de l'aplicació. El script Ulmanager s'encarrega també de gestionar els canvis d'escena en funció dels botons que premi l'usuari. Totes les escenes principals creades tenen la següent estructura:

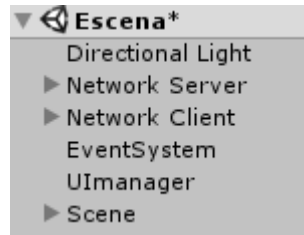


Figura 37. Estructura de l'escena amb la UI

Es mantenen els dos elements de comunicació: Servidor i Client. A més, s'afegeix el UIManager, que té assignat el codi que gestiona la interfície gràfica. Finalment hi ha l'objecte "Scene", que representa tots els elements de l'escena virtual.

7.8. Realitat Virtual

7.8.1. Aplicació VR

S'ha desenvolupat una aplicació anomenada VR per a demostrar els conceptes necessaris per al desenvolupament d'aplicacions de realitat virtual. Per a realitzar una aplicació d'aquest tipus reproduïda des d'un dispositiu Android, es parteix de l'estructura de la figura 37.

Si es vol crear una aplicació de realitat virtual, només cal fixar el tracker al dispositiu mòbil com a la figura 38 i assignar la seva posició i orientació a la càmera que visualitza l'escena. Al fer això, l'aplicació podrà detectar en tot moment on està posicionat el dispositiu i realitzar-ne un seguiment d'aquest. Per tant, la pantalla del smartphone fa de finestra al món virtual. Això crea una sensació de que l'usuari està movent-se realment dins l'escena virtual.

Dins del GameObject anomenat Scene de la figura 37, s'han d'introduir tots els elements virtuals que conformen l'escena.



Figura 38. Tracker fixat al smartphone utilitzat pel projecte

Per a definir l'escena demostrativa s'ha partit de l'escena on s'havia implementat el servidor, el client i en la que ja s'havia importat el plugin de SteamVR. A continuació, s'han afegit models 3D per a definir el món virtual que ha d'observar l'usuari. L'escena virtual és una sala ambientada en l'època medieval. S'ha tingut en compte que l'espai és limitat, ja que si l'usuari es fixa en la pantalla i no en l'entorn, podria xocar amb diversos elements. El resultat final és el que es mostra en la següent figura.



Figura 39. Aplicació de realitat virtual

A continuació es pot veure el punt de vista de l'usuari que executa l'aplicació, on es veu clarament el món virtual i l'entorn real.



Figura 40. Punt de vista de l'usuari

A la figura 40, es veu de fons el Centre de Realitat Virtual, mentre que al dispositiu es visualitza una realitat completament diferent. Aquesta escena demostra la quantitat de possibilitats que ofereix aquesta tecnologia pel que fa a les aplicacions de realitat virtual. A partir d'aquesta implementació, es pot definir qualsevol tipus d'escena amb els elements que es desitgi per a crear un nou món que es pot reproduir amb un qualsevol dispositiu mòbil.

7.9. Realitat Augmentada

7.9.1. Estat de l'art

La realitat augmentada és un camp que encara està en ple desenvolupament. Existeixen diferents dispositius per a visualitzar aquest tipus d'aplicacions com per exemple les HoloLens de Microsoft, unes ulleres que detecten l'entorn en 3D i projecten informació i objectes virtuals sobre aquest.



Figura 41. HoloLens de Microsoft

Aquest tipus de dispositius basats en ulleres, són els que proporcionen una millor experiència pel que fa a la qualitat de visualització. Tot i així, sembla que són encara molt cars per a l'usuari mig i els principals usuaris actualment són desenvolupadors d'aplicacions. El sector en el que han anat sorgint una gran quantitat de noves aplicacions és el dels dispositius mòbils, ja que la majoria de persones ja en tenen un.

La realitat augmentada per a dispositius mòbils va tenir un "boom" al juliol de 2016, quan va sortir al mercat l'aplicació de Pokemon GO. Aquesta va ser la primera aplicació de realitat virtual amb milions d'usuaris actius. L'aplicació es basa en els mapes de les ciutats per situar als usuaris, i segons la zona on es situïn, apareixen uns elements virtuals o altres a sobre el terra. En aquests cas, els objectes virtuals són pokemons, que són criatures basades en la sèrie d'animació homònima. Els usuaris poden anar caminant i visualitzant amb el smartphone el seu entorn per a veure si apareixen pokemons virtuals i així poder capturar-los.



Figura 42. Aplicació de AR Pokemon GO

Pel que fa al desenvolupament d'aplicacions de realitat augmentada amb Unity, existeix un plugin anomenat Vuforia. Aquest plugin permet detectar un patró pla i posicionar un element virtual a sobre d'aquest patró (figura 43). Els patrons són imatges que cal registrar en la seva web per a que siguin acceptats i el plugin les reconegui. És un sistema prou precís, tot i que depèn molt del tipus de patró que s'utilitzi.



Figura 43. Patró de Vuforia amb model 3D a sobre

Aquesta tecnologia té la limitació d'haver d'utilitzar patrons predefinits,. A més, si no es disposa del patró en el moment de l'execució, no es podrà posicionar l'objecte virtual a sobre. En aquest projecte s'ha estudiat una nova alternativa per al desenvolupament d'aplicacions de realitat augmentada amb Unity: utilitzant els trackers d'HTC.

7.9.2. Funcionament AR

La realitat augmentada (AR), és un tipus de realitat virtual que consisteix en superposar objectes o animacions generades per computador sobre una imatge a temps real que s'obté de la càmera del dispositiu que la reproduïx.

Es basa en dues capes: en la capa inferior es situa la imatge del dispositiu, i a la capa superior els objectes virtuals. Aquestes capes es veuen clarament en la figura 44.



Figura 44. Funcionament de la realitat augmentada

La combinació de les dues capes crea l'efecte de tenir objectes virtuals en la mateixa escena real. El principal problema que té aquesta tecnologia és la oclusió. Si un objecte es situa just davant de la càmera, sempre formarà part de la capa inferior i no es veurà mai per davant dels objectes virtuals. Actualment s'està intentant solucionar aquest problema amb algorismes d'intel·ligència artificial o amb càmeres de profunditat. Les càmeres de profunditat sí donen la informació 3D en l'espai. En aquest cas s'han utilitzat imatges 2D i per tant, el problema de la oclusió existeix.

7.9.3. Obtenció de la imatge del dispositiu

Com ja s'ha comentat en l'apartat 7.9.2, per a desenvolupar una aplicació de realitat augmentada, és necessari obtenir una imatge a temps real de l'entorn que representi la realitat. Per aconseguir accedir a la càmera s'ha creat el codi DeviceCamera.

Aquest codi s'encarrega d'accedir a la càmera del dispositiu, obtenir la imatge i assignar-la a la capa inferior.

Aquest codi s'ha d'assignar a la càmera de visualització de l'escena.

7.9.4. Implementació AR

Per a donar l'efecte de realitat augmentada, s'ha de crear un pla on es reproduïx les imatges obtingudes del dispositiu i superposar els objectes virtuals. Per fer això s'ha decidit crear un Canvas, que serà la capa inferior.

Es molt important configurar el Canvas com a "Screen Space – Camera" tal i com es veu a la figura 45.

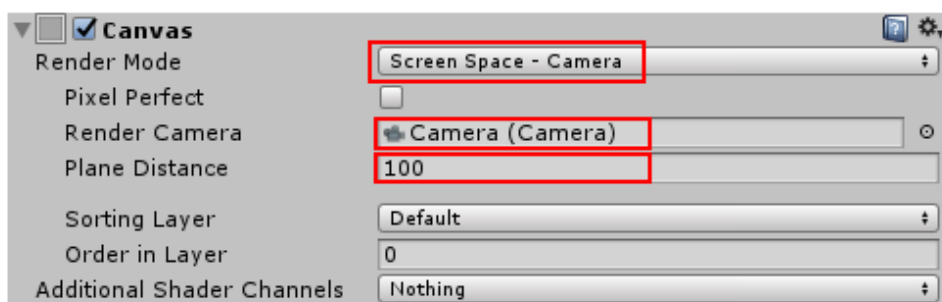


Figura 45. Configuració del Canvas

Com s'ha comentat al definir la UI, per defecte, el Canvas es situa sempre superposat a la imatge. Però existeixen altres opcions per a col·locar-lo de diverses maneres. El fet de definir-lo com a "Screen Space – Camera" permet donar-li profunditat a aquest pla i mantenir un offset respecte la càmera. Aquesta profunditat es pot escollir introduint el valor de la distància. També s'ha de seleccionar quina és la càmera que visualitza l'escena.

Al Canvas cal afegir-li un element de UI anomenat "raw image", que no és més que una imatge en blanc a la que se li pot assignar el valor de qualsevol textura. En aquest cas, se li ha assignat com a textura la imatge de la càmera del mòbil. Al "raw image" del Canvas se li ha d'afegir un element anomenat aspect ratio, que indica la proporció entre altura i amplada de la imatge. S'ha de seleccionar la opció "Envelope Parent", que significa que se li assignarà el ratio del seu element "pare", és a dir, el del Canvas.

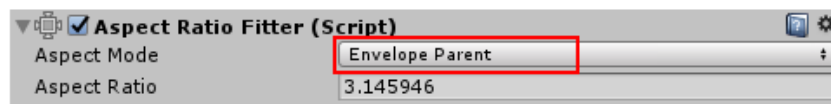


Figura 46. Configuració del rati d'aspecte

El Canvas s'adaptarà a la pantalla del dispositiu i això permet crear aplicacions molt versàtils

que es poden reproduir en pantalles de qualsevol mida i tipus.

Finalment, a les opcions de la càmera, s'ha de modificar la distància que és capaç de renderitzar. És a dir, tots els elements que estiguin situats en una posició més llunyana de la indicada, no es visualitzaran. Aquesta distància ha de ser superior a la que es col·loqui el Canvas, ja que sinó aquest no es podria visualitzar.

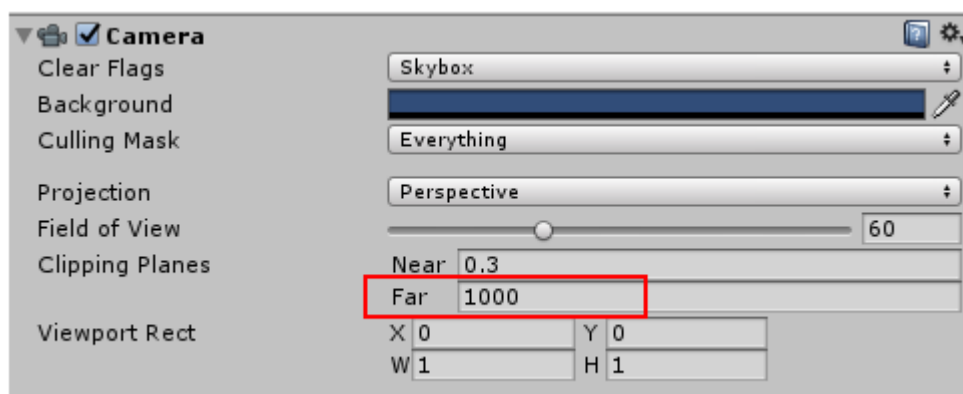


Figura 47. Distància de renderitzat

A la següent figura es mostra la càmera a l'esquerra, amb els tres eixos x,y i z. Amb línies grises es representa la zona de visualització de la càmera. En aquest Canvas de color blanc, es visualitzarà la imatge que s'obté de la càmera del dispositiu.

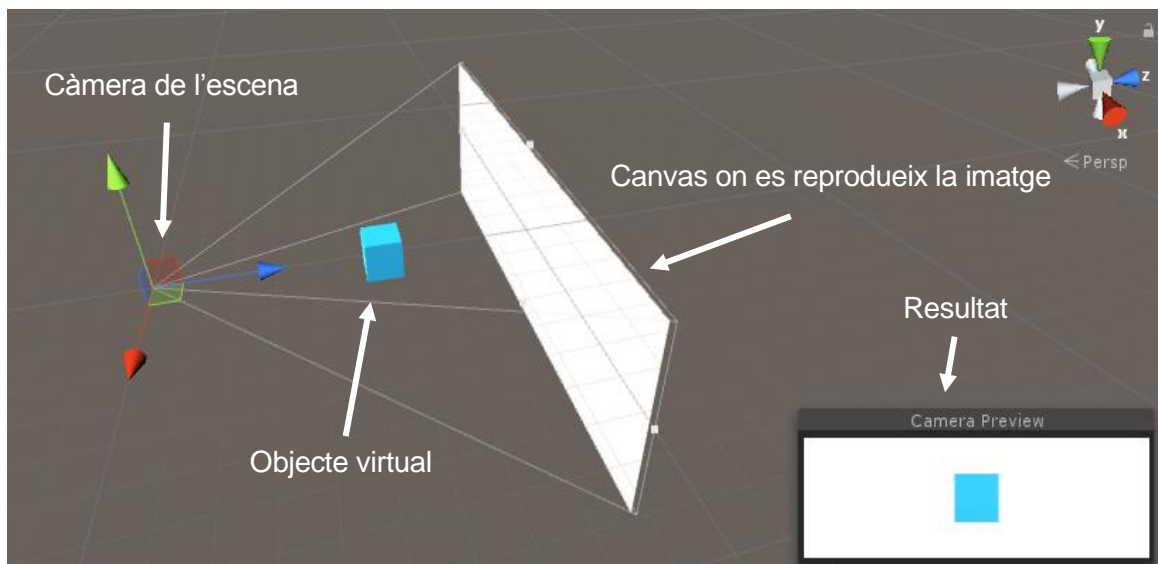


Figura 48. Implementació de realitat augmentada amb Unity

Quan ja s'ha definit la distància a la que es situa el pla, només cal posicionar objectes entre la càmera i la capa inferior. Com es veu a la figura 48, s'ha col·locat un cub de color blau enmig, i a la part inferior a la dreta, es representa el que capta la càmera, que és el que es

veuria en el dispositiu mòbil. En aquesta petita finestra es veu una cara del cub i el fons de color blanc.

L'escena demostrativa de realitat augmentada té la següent estructura.

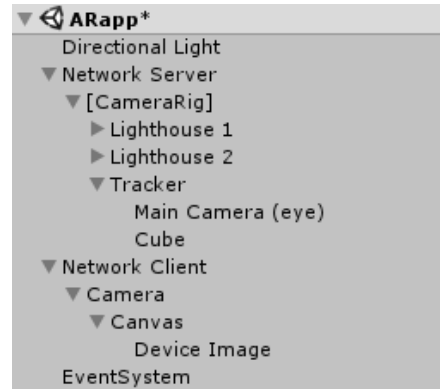


Figura 49. Estructura de l'escena de realitat augmentada

Com es pot observar, apareixen tots els elements comentats, que pegen del “GameObject” del client. Quan es reproduïxi l'aplicació des del PC com a servidor no serà necessari obtenir dades de cap càmera i per tant es desactivarà el Network Client i tots els elements que pegen d'ell.

7.9.5. Aplicació AR

L'aplicació AR té com a objectiu demostrar el funcionament de la realitat augmentada amb els dispositius d'HTC. S'han utilitzant els scripts creats per a les comunicacions amb el sistema Client/Servidor per obtenir la posició i orientació del tracker des del smartphone. Amb el procediment descrit en l'apartat anterior, es pot crear una escena de realitat augmentada i visualitzar-la des del mòbil.

Només cal col·locar els objectes virtuals entre la càmera i el Canvas com el cub de la figura 48 de l'apartat anterior i ja es pot visualitzar-los, donant la sensació de que aquests objectes estan posicionats en l'entorn real.



Figura 50. Balena de realitat augmentada

La figura 50 és una imatge de l'aplicació AR, en la que s'ha col·locat una balena al centre de l'habitació. Aquesta balena té una animació assignada que la fa moure, donant la sensació de que forma part de l'entorn.

Es pot crear tot tipus d'aplicacions segons els requeriments específics, introduint diversos models 3D com per exemple la balena amb les animacions pertinents.

El problema principal d'aquesta implementació és que l'usuari és un mer espectador i no té cap manera d'interactuar amb els objectes virtuals. Per aquest motiu, s'ha decidit crear una aplicació més interactiva que s'explica en el següent apartat.

7.10. Selecció i posicionament d'objectes

És interessant aprofundir en les possibilitats que ofereixen els trackers, ja que són uns dispositius molt versàtils i són els que tenen un preu més econòmic. Cada tracker permet obtenir només una orientació i una posició. És de gran interès la possibilitat d'obtenir dades de més d'un tracker simultàniament en aplicacions de realitat augmentada, ja que un tracker s'ha de fer servir per a posicionar càmera i cal fixar-lo al dispositiu mòbil amb el que es visualitzarà l'aplicació. Al afegir-ne un de nou, apareix l'opció d'interactuar amb l'aplicació, millorant així la sensació d'immersió.

7.10.1. Aplicació ARposition: selecció i posicionament d'objectes

L'escena ARposition és una demostració de les possibilitats que ofereix el projecte desenvolupat. A partir dels codis creats durant el projecte i les opcions que s'han implementat, es poden fer servir de punt de partida per a realitzar qualsevol aplicació de realitat augmentada. Aquesta aplicació, pretén crear una escena de realitat augmentada personalitzada per a l'usuari. Primerament, de la mateixa manera que en aplicacions anteriors, apareix una escena amb una UI on s'ha d'introduir la IP del servidor per poder connectar-se i rebre les dades dels trackers. Una vegada es coneix la IP, s'executa l'escena principal. Aquesta escena, de fons, té la imatge obtinguda de la càmera del dispositiu tal i com s'ha definit en l'aplicació anterior anomenada "AR". Les dades del primer tracker s'assignen a la càmera de l'escena i representen el smartphone de l'usuari, permetent que l'usuari es desplaci dins l'escena.

Una vegada s'ha seleccionat la IP, apareix un menú superposat a la imatge on es pot seleccionar el model 3D que es desitgi afegir a l'escena. Quan ja s'ha seleccionat el model, es procedeix a posicionar-lo en algun punt de l'entorn mitjançant un segon tracker. Finalment, també s'ha afegit la opció d'escalar el model introduït per a que aquest tingui la mida desitjada per a l'usuari. El comportament d'aquest menú es gestiona amb el codi desenvolupat anomenat "Ulmanager".

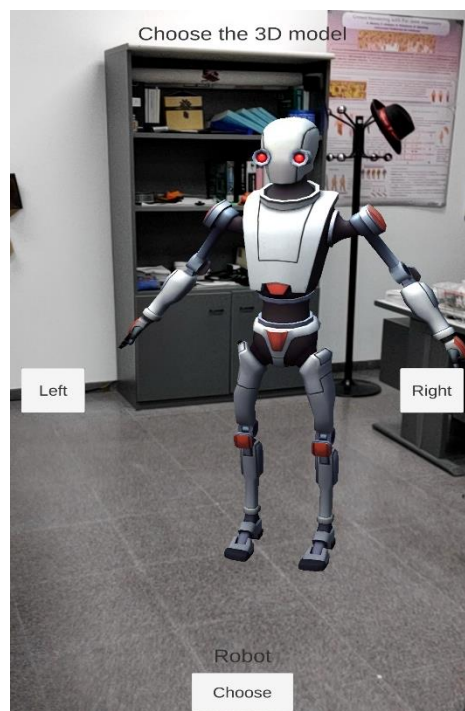


Figura 51. Menú de selecció d'elements 3D

En la figura 51, es pot veure el menú creat per a seleccionar els objectes. Aquest permet anar visualitzant els diferents models i seleccionar el que es desitgi posicionar.

Quan l'objecte té la mida i la posició desitjada, es pot fixar en un punt a l'espai. L'usuari pot anar col·locant els objectes escollint la posició exacta on vol que es mantinguin. Això permet interactuar amb l'escena creant-ne una de pròpia per a l'aplicació que desitgi l'usuari. És una manera simple, eficient i de baix cost, de visualitzar elements virtuals sobre una escena real per veure quina aparença tindrien sense haver de tenir-los físicament.

Per a gestionar les dades de dos trackers, s'ha ampliat la classe "missatge" definida anteriorment. En aquest cas, com l'objectiu és utilitzar dos trackers simultàniament, s'han afegit els atributs de posició i orientació d'un segon tracker. Si es volguessin utilitzar més de dos, només caldria afegir dos atributs (posició i orientació) per a cadascun dels trackers que es vulguin gestionar.

La classe queda de la següent manera:

```
//Creem la classe per definir els atributs que volem enviar
public class CameraMessage : MessageBase
{
    //Decidim els atributs que tindra la classe creada

    //tracker 1
    public Quaternion tracker_orientation;
    public Vector3 tracker_position;

    //tracker 2
    public Quaternion tracker_orientation2;
    public Vector3 tracker_position2;
}
```

Figura 52. Classe per a dades de dos trackers

El servidor enviarà contínuament les dades dels dos trackers. El client, al estar connectat al servidor, podrà llegir aquestes dades i guardar-les. Una vegada les ha obtingut, ja es poden assignar a la càmera i a l'objecte que es vulgui posicionar.

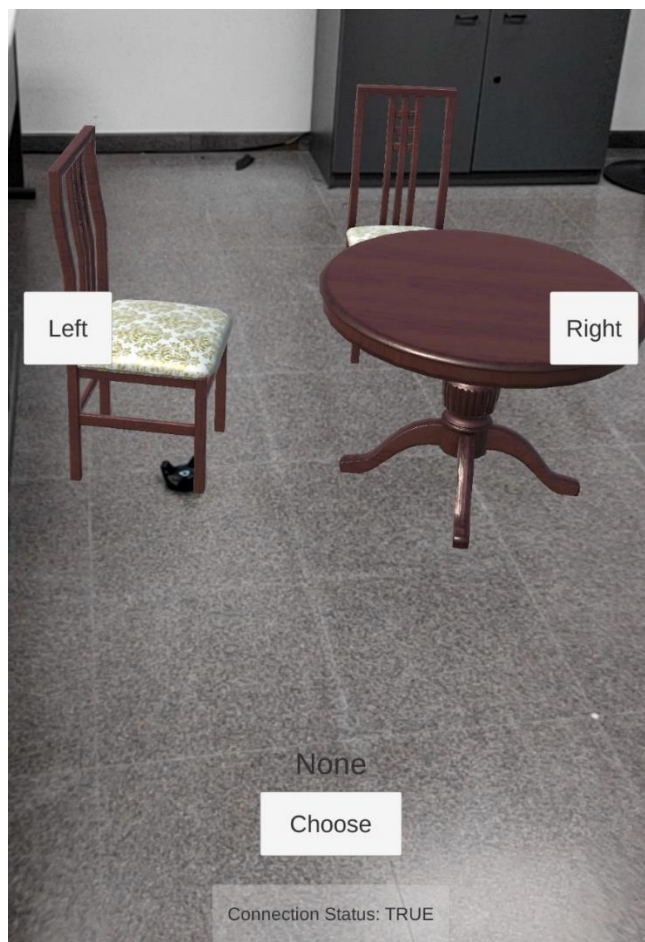


Figura 53. Introducció d'elements a l'escena

A la figura 53 , es pot veure una escena en la que s'han introduït una sèrie d'objectes virtuals. S'han afegit una taula i dues cadires a sobre el terra, donant la sensació de que formen part de l'entorn real.

Amb els conceptes definits es poden crear tot tipus d'aplicacions que necessitin de la gestió de dos trackers simultàniament i el posicionament d'objectes virtuals.

8. Aplicacions multiusuari

Coneixent la metodologia per a gestionar les dades de dos trackers a la vegada, es pot implementar la possibilitat de que dues o més persones visualitzin la mateixa escena simultàniament. A cadascuna se li assigna un tracker i aquests es fan servir respectivament com a càmeres dels usuaris dins l'escena. L'aplicació demostrativa que presenta aquest concepte s'anomena Multiuser.

Per gestionar les dades, s'ha creat una interfície d'usuari que, una vegada el smartphone ja s'ha connectat al PC per obtenir dades, permet escollir de quin tracker específic es volen obtenir. En aquest cas s'han utilitzat dos, i cal seleccionar l'índex del primer o del segon com es mostra a la figura 54. Cadascun dels usuaris ha d'escollir un tracker diferent.



Figura 54. UI per a la selecció del tracker

Aquest concepte obre un nou ventall de possibilitats, ja que, combinat amb el l'aplicació anterior "ARposition" on es posicionaven objectes, permetria la interacció amb l'entorn de diversos usuaris. A més, cada usuari podria llegir també les dades de la posició dels altres usuaris i això els permetria la interacció entre ells.

Un cas d'ús per a aquest concepte és la possibilitat de que es connectin diversos usuaris a la vegada, definint-ne un d'ells com a usuari principal com amb la Powerwall. Aquest usuari podria gestionar l'escena des del seu smartphone per visualitzar els models que li interessi

mentre els altres es situen a l'escena com a espectadors. Aquest tipus d'aplicació és molt útil en l'àmbit comercial d'una empresa. Un venedor podria mostrar als clients el resultat del seu producte de manera virtual i el client podria moure's amb llibertat dins l'escena per visualitzar les parts que més li interessin.

En el sector de la formació, aquest concepte permet definir un usuari principal com a mentor, mentre que els alumnes serien els visualitzadors. Això permetria mostrar amb facilitat els processos o models en 3D. Això seria de gran utilitat en l'estudi de l'anatomia o de la biologia per exemple.

En el cas de la realitat virtual, en el que no s'obté una imatge real de l'entorn, es podrien visualitzar avatars 3D dels altres usuaris per a saber en quin lloc de l'escena estan posicionats. Això evitaria col·lisions entre usuaris i fomentaria la interacció entre ells.

9. Cas d'ús: instruccions de muntatge

9.1. Motivació

És ben sabut, que per a moltes persones, el fet d'haver de seguir instruccions si no s'han definit amb molta claredat pot arribar a ser un problema, sobretot en el cas d'un muntatge. És per això que s'ha creat una aplicació anomenada ARinstructions que permet ajudar a entendre instruccions i facilita la feina a l'usuari per a que executi a la perfecció les diferents instruccions.

9.2. Objectiu

L'objectiu de l'aplicació és mostrar pas per pas, quines són les tasques que s'han de realitzar en un muntatge. Com a exemple s'ha implementat el muntatge d'un robot industrial manipulador (figura 55). L'aplicació ha de mostrar les instruccions escrites i també representar-les de forma visual. També ha de permetre continuar amb la següent instrucció i tornar a la instrucció anterior mitjançant la interfície de l'usuari. Això es realitza amb els botons tàctils de "previous" i "next".

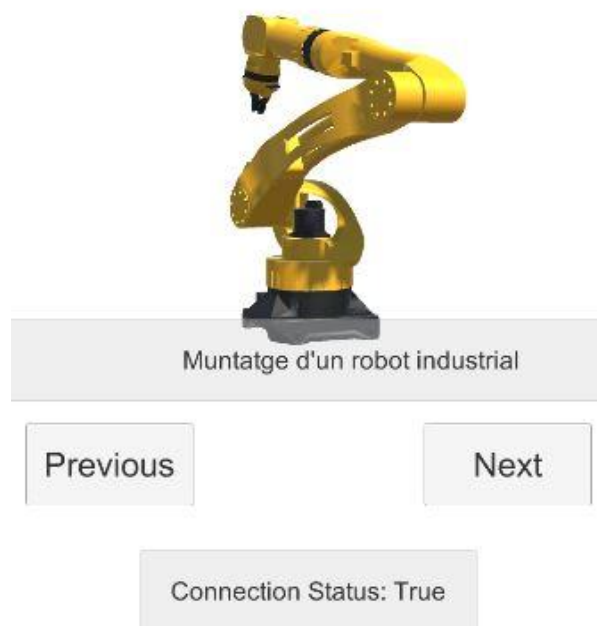


Figura 55. Aplicació pel muntatge d'un robot industrial

9.3. Aplicació ARinstructions

L'aplicació ARinstruccions, basada en aplicacions creades anteriorment, disposa d'una UI on l'usuari pot seleccionar quina instrucció visualitzar. La idea és que comenci per la primera i vagi avançant a la següent mentre es realitza el muntatge real en paral·lel. En el cas de que aparegui algun error, es pot tornar a visualitzar instruccions anteriors per trobar el problema i solucionar-lo. A la figura 55, es mostra la interfície gràfica de l'aplicació.

S'ha creat un codi que defineix quins són els passos per a realitzar el muntatge. El codi gestiona quines instruccions han d'aparèixer en el panell on es visualitza el text i també va afegint les noves peces virtuals per a cada pas. Així l'usuari pot anar seguint les instruccions i visualitzant el resultat simultàniament. La figura 56 mostra les dades que demana el codi per a poder crear l'aplicació. A la part superior s'afegeixen les peces individuals en ordre de muntatge. A la part inferior anomenada "instructions", s'escriuen les instruccions de cada pas, que quedaran guardades dins el codi com a variables del tipus "string" i s'aniran mostrant per pantalla a mesura que l'usuari premi el botó de "next" o "previous".

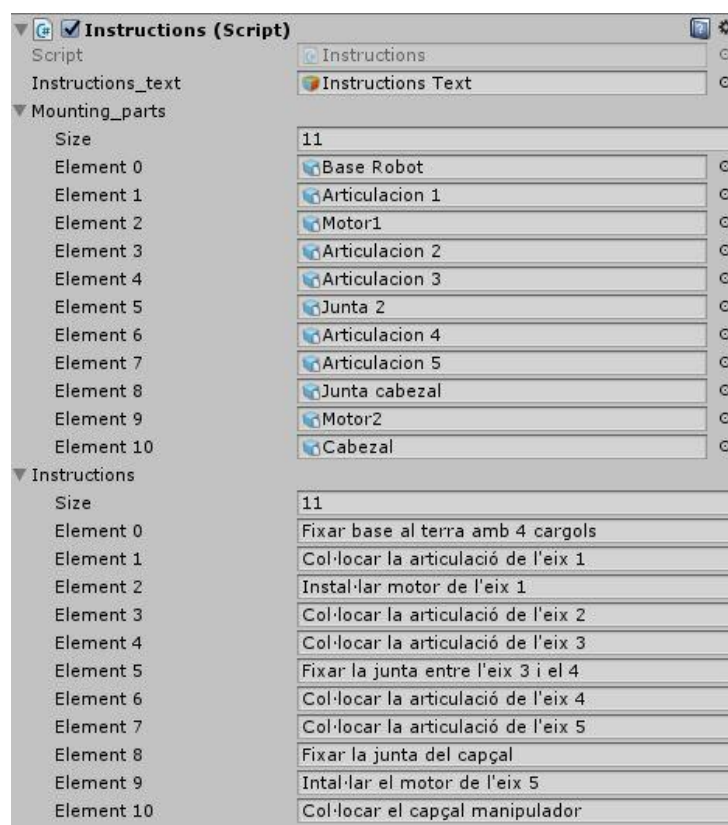


Figura 56. Variables a introduir en el codi de muntatge

A la figura 57, a la pàgina següent, es pot veure una captura de pantalla de l'aplicació. A la imatge, encara falta instal·lar el motor i col·locar el capçal manipulador.



Figura 57. Muntatge d'un robot industrial

9.4. Aplicacions pràctiques

Aquest concepte es pot aplicar en el món industrial. En qualsevol cadena de muntatge, aquesta aplicació permet que l'operari es formi d'una manera més ràpida i eficient. També li dona la seguretat de poder consultar en qualsevol moment quin és el següent pas de forma senzilla en cas de dubte.

El mateix es pot dir de plantes de producció, on sovint apareixen peces defectuoses degut a errors humans causat per dubtes pel que fa al procés en sí. L'operari podria visualitzar en una aplicació com aquesta quin és el següent procés a realitzar, quin objecte ha de moure a quina màquina o quin polsador ha de prémer per a activar el següent procés per exemple. Això reduiria el nombre de peces rebutjades i milloraria l'eficiència de la planta productiva.

Finalment, pel que fa als manuals d'instruccions, l'ús d'aquest tipus de tecnologia permetria seguir els passos d'una manera més simple. Per exemple, en el cas de la famosa empresa de mobles IKEA, el fet de que el client disposés d'aquesta tecnologia li permetria crear manuals d'instruccions basats en realitat augmentada. L'usuari podria visualitzar en la seva pròpia habitació com ha d'anar muntant el moble que ha adquirit i visualitzar-ho des de qualsevol angle per entendre com s'ha de fer.

10. Planificació del projecte

10.1. Gestió del temps

La gestió del temps s'ha realitzat mitjançant una fulla de càlcul, on s'han introduït al llarg del temps dades sobre com el projecte va avançant fins assolir els objectius finals.

10.2. Diagrama de Gantt

El projecte es pot dividir en 9 activitats depenent de la naturalesa de les tasques realitzades:

- A1. Introducció als dispositius HTC
 - Durant aquesta activitat, s'ha desenvolupat una escena on s'utilitzen les dades de posicionament i tracking que proporcionen els diferents dispositius per entendre el seu funcionament i analitzar com es gestionen aquestes dades per a poder utilitzar-les posteriorment per al desenvolupament d'aplicacions.
- A2. Desenvolupament d'aplicacions sense el HMD
 - Durant aquesta activitat s'ha estudiat la possibilitat de desconnectar el HMD per a poder desenvolupar i reproduir aplicacions sense haver de connectar-lo. S'ha modificat el codi de SteamVR per a poder definir l'habitació sense connectar el casc i s'ha estudiat l'ordre d'assignació dels índex dels dispositius.
- A3. Comunicacions entre dispositius
 - Durant aquesta activitat, s'han analitzat les capacitats dels dispositius mòbils per a la gestió de dades dels dispositius HTC i s'ha desenvolupat un sistema del tipus Servidor/Client mitjançant wifi per a la connexió entre dispositius.
- A4. Realitat Virtual
 - Durant aquesta activitat s'ha estudiat la possibilitat de desenvolupar aplicacions de realitat virtual amb el sistema de comunicacions definit. S'han assignat les dades de posicionament i tracking a la càmera de l'escena per a que l'usuari pugui desplaçar-se dins l'escena virtual. També s'ha definit la metodologia per

a la creació d'aplicacions per Android i la manera d'implementar-les amb Unity.

- A5. Realitat augmentada
 - Durant aquesta activitat s'ha estudiat com és el funcionament de la realitat augmentada i com implementar-la amb els dispositius HTC, introduint objectes virtuals superposats en una escena real.
- A6. Aplicació de posicionament
 - Durant aquesta activitat s'ha desenvolupat una aplicació de realitat augmentada on l'usuari pot posicionar objectes virtuals en una escena real mitjançant els trackers. A més, s'ha definit com gestionar les dades de més d'un tracker simultàniament.
- A7. Multiusuari
 - Durant aquesta activitat s'ha desenvolupat una aplicació per gestionar les dades de més d'un usuari al mateix temps, permetent a més d'una persona visualitzar la mateixa escena de realitat augmentada.
- A8. Escena de muntatge
 - Durant aquesta activitat s'ha desenvolupat una aplicació basada en tots els coneixements adquirits. L'aplicació és un cas d'ús pràctic d'aquesta tecnologia, que pretén facilitar la comprensió d'una sèrie d'instruccions per al muntatge d'un ensamblatge mitjançant la realitat augmentada.
- A9. Documentació
 - Durant aquesta activitat s'ha documentat tota la informació recopilada sobre el funcionament de la tecnologia tractada, les aplicacions actuals i les escenes desenvolupades durant el projecte.

Aquestes activitats es subdivideixen alhora en tasques que s'han dut a terme per a l'obtenció de l'òptim resultat final. Per a resumir gràficament en ordre cronològic totes les tasques relacionades amb el projecte, s'ha realitzat un diagrama de Gantt que representa l'inici i el fi de cada tasca respecte el temps (figura 58).

Les activitats han estat realitzades durant els mesos que ha durat el projecte en el període comprés entre el 20/09/2018 i el 17/01/2019. Aquestes tasques s'han representat gràficament mitjançant un diagrama de Gantt.

El diagrama de Gantt (figura 58 a la pàgina 61) representa cronològicament l'inici i fi de totes les tasques. Aquest diagrama mostra per setmanes el seguit d'activitats realitzades i la seva durada. El fet de planificar prèviament i d'anar afegint les noves activitats, ha facilitat molt la gestió del temps i s'ha pogut analitzar en tot moment la situació del projecte per assegurar que s'assoleixen tots els objectius abans de que finalitzi la fita. El projecte s'ha realitzat en els cinc mesos que dura el semestre. A la següent figura es pot veure amb més detall les activitats concretes que s'han definit.

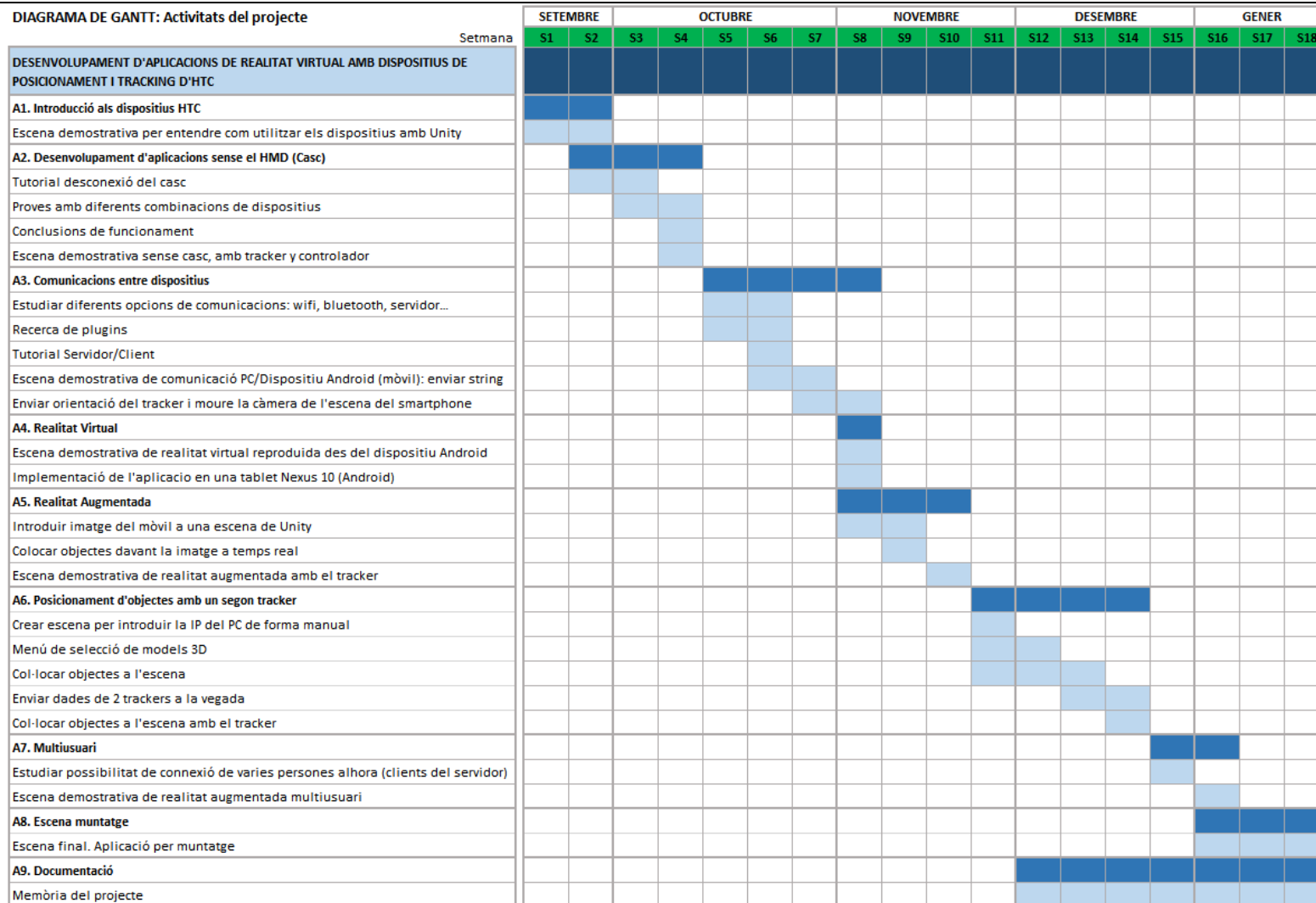


Figura 58. Diagrama de Gantt

11. Pressupost

Aquest no és un projecte de gran envergadura que requereixi de grans infraestructures ni inversions significatives. Al cap i a la fi, es tracta de desenvolupament de software i, per tant, econòmicament, l'impacte és mínim. Sí és cert, però, que diversos recursos són imprescindibles per realitzar-lo. Una de les motivacions més importants d'aquest projecte és el baix cost de les eines requerides per dur-lo a terme.

Aquests dispositius no són d'utilitat si no es disposa d'un computador amb els requeriments mínims que demana. L'ordinador utilitzat ha estat un PC valorat en 1000 euros.

Per executar les aplicacions i fer proves, s'han fet servir dos smartphones. Un Pocophone f1 valorat en 300 euros i un Samsung Galaxy A3 2016 valorat en 280.

Per altra banda, el motor de jocs Unity disposa de diferents versions, segons les necessitats del client. Per aquest projecte s'ha utilitzat la versió Personal Unity 2017.3.1f1 (64-bit), que es la més simple i es pot adquirir gratuïtament. Aquesta versió té les seves limitacions, com per exemple, el límit de beneficis de 100.000 \$ si es comercialitza una aplicació creada amb aquest software. En aquest cas no és un problema, ja que aquest projecte no es comercial, sinó que la seva finalitat és merament acadèmica.

També s'ha utilitzat el software SteamVR que es pot adquirir de forma gratuïta.

Un fet a destacar és la possibilitat - des del llançament de la versió Unity 5 el 3 de març del 2015, que va donar la possibilitat als usuaris de versions Personal (gratuïtes) - d'utilitzar plugins externs a Unity. Anteriorment a aquest esdeveniment, per utilitzar un plugin era necessari disposar d'una versió Pro, valorada en 125 \$ al mes per usuari.

L'últim software necessari ha estat el paquet Office de Microsoft, necessari per poder crear el present document i les taules i gràfics propis que clarifiquen l'explicació i exposició del projecte. La llicència d'aquest software està valorada en 300 €.

Per acabar s'ha calculat el preu de l'enginyer encarregat d'executar el projecte, valorant el seu lloc de treball com a enginyer i programador en 20 € l'hora. A més, s'ha comptabilitzat el cost del director del projecte valorat en 30 € l'hora.

La taula resum, a continuació (Taula 3), conté les dades dels costos totals que els requeriments del projecte generen.

Taula 3. Pressupost del projecte

	Unitat	Preu unitari	Hores	Cost (€)
Personal Unity 2017.3.1f1	1	-	-	-
Software SteamVR	1	-	-	-
Pocophone f1	1	300	-	300,00 €
Samsung Galaxy A3 2016	1	280	-	280,00 €
Kit dispositius HTC	1	500	-	500,00 €
Tracker	2	100	-	200,00 €
PC	1	1000	-	1.000,00 €
Llicències informàtiques (Office)	1	300	-	300,00 €
Hores de treball d'enginyer	h	20	350	7.000,00 €
Director de projecte	h	30	50	1.500,00 €
Total				11.080,00 €

Per tant, el cost final del projecte és de **11.080 €**.



Conclusions

La realitat virtual és una tecnologia relativament nova, que està guanyant popularitat els darrers anys. És per això que és molt interessant aprofundir en aquest tema i desenvolupar un projecte basat en les possibilitats que aquesta tecnologia proporciona.

Durant el projecte s'ha analitzat el funcionament dels dispositius d'HTC i com gestionar les dades que proporciona. A partir d'aquestes dades, s'ha definit una metodologia per a la creació d'escenes virtuals.

S'ha proposat una nova alternativa a les tecnologies actuals per a realitzar aplicacions de realitat augmentada. Aquesta alternativa posiciona a l'usuari amb un tracker i es visualitza amb dispositius mòbils.

A més, s'ha aprofundit en la possibilitat de gestionar diversos dispositius simultàniament, fet que afegeix noves opcions a l'hora d'interactuar amb l'entorn o amb altres usuaris.

Finalment s'ha creat una aplicació com a exemple del que es pot arribar a fer amb els coneixements adquirits durant el projecte.

En conjunt, es pot dir que s'ha creat una base sòlida per a poder desenvolupar aplicacions de realitat virtual i augmentada mitjançant els dispositius de posicionament i tracking d'HTC i, per tant, es considera assolit l'objectiu inicial del projecte.

És qüestió de temps que la tecnologia 5G guanyi protagonisme en el món dels dispositius mòbils. Això permetrà augmentar la velocitat de les xarxes, obtenir una menor latència i millorar la connectivitat. Aquesta tecnologia suposa un canvi en el sector de la realitat virtual, ja que els nous dispositius que es desenvolupin, permetran executar aplicacions més complexes sense la necessitat d'utilitzar ordinadors potents d'un cost elevat. En aquest document s'exposen algunes de les possibilitats que aquest canvi pot suposar.

Agraïments

Voldria agrair al tutor del projecte, Toni Susin, per haver-me donat l'oportunitat de tornar a realitzar un projecte relacionat amb el món de la realitat virtual. Per ajudar-me a entendre conceptes relacionats amb el projecte en moments de frustració i per la seva disponibilitat i ràpida resposta quan ha sorgit algun dubte o problema.

També vull mencionar a en Jordi Moyés, per oferir-me el seu material i espai de treball al Centre de Realitat Virtual de la FME. Gràcies per la seva confiança i disponibilitat.

Finalment agrair a la meua família i amics, que m'han donat suport en tot moment durant l'elaboració del projecte.

Bibliografia

Referències bibliogràfiques

- [1] RAE. Definició de realitat virtual.

[<https://dle.rae.es/srv/fetch?id=VH7cofQ>, 3 gener 2019]

- [2] Sean Buckley. This is how Valve's amazing Lighthouse tracking Technology Works.

[<https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768>, 4 octubre 2018]

- [3] Reid Wender. *SteamVR tracking without an HMD*

[<http://help.triadsemi.com/steamvr-tracking/steamvr-tracking-without-an-hmd>, 20 setembre de 2018].

- [4] Ari Laiho. Visualització d'escenes de Realitat Virtual amb ulleres HTC Vive amb interacción (2017)

[<https://mat-web.upc.edu/people/toni.susin/images/projects2/Ari%20-%20HTC.pdf>, 24 setembre 2018]

- [5] Isabel Valenzuela. ¿Es Bluetooth más veloz que Wifi?

[<https://www.vix.com/es/btg/tech/13523/es-bluetooth-mas-veloz-que-wifi>, 17 octubre 2018].

- [6] Tutorial Unity Android

[<https://unity3d.com/es/learn/tutorials/topics/mobile-touch/building-your-unity-game-android-device-testing>, 19 octubre 2018]

- [7] Solvetic. Cómo activar depuración USB Pocophone F1

[<https://www.solvetic.com/tutoriales/article/6487-como-activar-depuracion-usb-pocophone-f1/>, 19 octubre 2018]

Bibliografia complementària

Unity Manual. Multiplayer and Networking.

[<https://docs.unity3d.com/Manual/UNet.html>]

Manual d'instruccions HTC Vive

[https://dl4.htc.com/Web_materials/Manual/Vive/Vive_User_Guide.pdf?_ga=2.163210812.1061259425.1546426272-890888918.1544611619]

Plataforma de models 3D

[<https://free3d.com/3d-models/obj>]